Efficient use of a Protein Structure Annotation Database

Application to packing analysis

DISSERTATION

zur Erlangung des akademischen Grades
doctor rerum naturalium
(Dr. rer. nat.)
im Fach Biologie

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät I
Humboldt-Universität zu Berlin

von
Herr Dipl.-Biochem. Kristian Rother
geboren am 11.4.1977 in Berlin

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Christoph Markschies

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät I:
Prof. Thomas Buckhout, PhD

Gutachter:

1. Prof. Dr. Cornelius Frömmel
2. Prof. Dr. Ulf Leser
3. Janusz M. Bujnicki, PhD, DHabil

eingereicht am:	21. Mai 2006
Tag der mündlichen Prüfung:	20. September 2006

Abstract

In this work, a multitude of data on structure and function of proteins is compiled and subsequently applied to the analysis of atomic packing. Structural analyses often require specific protein datasets, based on certain properties of the proteins, such as sequence features, protein folds, or resolution. Compiling such sets using current web resources is tedious because the necessary data are spread over many different databases. To facilitate this task, Columba, an integrated database containing annotation of protein structures was created. Columba integrates sixteen databases, including PDB, KEGG, Swiss-Prot, CATH, SCOP, the Gene Ontology, and ENZYME.

The data in Columba revealed that two thirds of the structures in the PDB database are annotated by many other databases. The remaining third is poorly annotated, partially because the according structures have only recently been published, and partially because they are non-protein structures.

The Columba database can be searched by a data source-specific web interface at *www.columba-db.de*. Users can thus quickly select PDB entries of proteins that match the desired criteria. Rules for creating datasets of proteins efficiently have been derived.

These rules were applied to create datasets for analyzing the packing of proteins. Packing analysis measures how much space there is between atoms. This indicates regions where a high local mobility of the structure is required, and errors in the structure. In a reference dataset, a high number of atom-sized cavities was found in a region near the protein surface. In a transmembrane protein dataset, these cavities frequently locate in channels and transporters that undergo conformational changes. A dataset of ligands and coenzymes bound to proteins was packed as least as tightly as the reference data. By these results, several contradictions in the literature have been resolved.

Keywords:
protein structure, databases, data integration, data quality, annotation, protein packing

Zusammenfassung

Im Rahmen dieser Arbeit wird eine Vielzahl von Daten zur Struktur und Funktion von Proteinen gesammelt. Anschließend wird in strukturellen Daten die atomare Packungsdichte untersucht. Untersuchungen an Strukturen benötigen oftmals maßgeschneiderte Datensätze von Proteinen. Kriterien für die Auswahl einzelner Proteine sind z.B. Eigenschaften der Sequenzen, die Faltung oder die Auflösung einer Struktur. Solche Datensätze mit den im Netz verfügbaren Mitteln herzustellen ist mühselig, da die notwendigen Daten über viele Datenbanken verteilt liegen. Um diese Aufgabe zu vereinfachen, wurde Columba, eine integrierte Datenbank zur Annotation von Proteinstrukturen, geschaffen. Columba integriert insgesamt sechzehn Datenbanken, darunter u.a. die PDB, KEGG, Swiss-Prot, CATH, SCOP, die Gene Ontology und ENZYME.

Von den in Columba enthaltenen Strukturen der PDB sind zwei Drittel durch viele andere Datenbanken annotiert. Zum verbliebenen Drittel gibt es nur wenige zusätzliche Angaben, teils da die entsprechenden Strukturen erst seit kurzem in der PDB sind, teils da es gar keine richtigen Proteine sind.

Die Datenbank kann über eine Web-Oberfläche unter *www.columba-db.de* spezifisch für einzelne Quelldatenbanken durchsucht werden. Ein Benutzer kann sich auf diese Weise schnell einen Datensatz von Strukturen aus der PDB zusammenstellen, welche den gewählten Anforderungen entsprechen. Es wurden Regeln aufgestellt, mit denen Datensätze effizient erstellt werden können.

Diese Regeln wurden angewandt, um Datensätze zur Analyse der Packungsdichte von Proteinen zu erstellen. Die Packungsanalyse quantifiziert den Raum zwischen Atomen, und kann Regionen finden, in welchen eine hohe lokale Beweglichkeit vorliegt oder welche Fehler in der Struktur beinhalten. In einem Referenzdatensatz wurde so eine große Zahl von atomgroßen Höhlungen dicht unterhalb der Proteinoberfläche gefunden. In Transmembrandomänen treten diese Höhlungen besonders häufig in Kanal- und Transportproteinen auf, welche Konformationsänderungen vollführen. In proteingebundenen Liganden und Coenzymen wurde eine zu den Referenzdaten ähnliche Packungsdichte beobachtet. Mit diesen Ergebnissen konnten mehrere Widersprüche in der Fachliteratur ausgeräumt werden.

Schlagwörter:
Proteinstruktur, Datenbanken, Datenintegration, Datenqualität, Annotation, Packungsdichte

Acknowledgements

I would like to thank the following persons for their support during my graduation studies: Ulf Leser for continuous supervision of the Columba project. Robert Preißner for many helpful advice, Silke Trissl for her engagement in the Columba Web interface and uncounted bugfixes in Columba, Raphael Bauer for writing and unscrambling all BioPerl-related stuff, Heiko Müller for advice in creating the data model, Stefan Günther for writing the CATH module and analyzing DNA-protein interfaces, Elke Michalsky for mathematical advice and providing data on ligands from the PDB, Philipp Hussels for implementing the XML interface, Thomas Steinke for maintaining the Columba webserver, Patrick May and Ina Koch for providing PTGL data, Eike Staub and Antje Krause for providing SYSTERS data, Bingyu Zhu for inspiring the Columba database name, Lindy-Lynn Bright for advice on my English, Peter Hildebrand for his patience and martial arts entertainment, Pico for remaining calm most of the time, Björn Peters for advice on general aspects of scientific research, the whole AG Proteinstrukturtheorie, the Open-Source community for most of the software i used, and all the love and trust from Nils Goldmann and my family. Special acknowledgements go to Prof. Frömmel, who never hesitated in providing me with the most challenging (and thereby interesting) questions.

This project was supported by the German Ministry of Education and Research (BMBF), grant no. 0312705B.

Contents

1 Preface: knowledge, its conservation and mining

Knowledge consists of information and the ability to use it intentionally. In the *information hierarchy* model used in knowledge management, it follows data and information and precedes wisdom. While people in general should strive to reach wisdom, scientists acquire knowledge most of their time. Based on this, they try to create new knowledge not known before. By this method, a tradition of passing knowledge from scientist to scientist has emerged, as has been common among craftsmen, teachers, masters of art and parents for many centuries.

In recent years, the media used for passing information in science have changed. By the advance of computer technology, storing and preserving huge amounts of information without loss is no longer a problem. But information and knowledge are not equivalent, and this discrepancy becomes especially imminent as it comes to biological knowledge. By the efforts of the genome projects and many follow-up initiatives, impressive and ever more impressive numbers of nucleotides, genes, sequences and the like are being piled up. But what do we really know? The increasing volume of both databases and publications will make it more difficult to find the required information in a specific research task. There might exist heaps of useful facts within the data that cannot be found, because they are not linked by other data in a particular context. This carries the risk of false conclusions because it can be interpreted in a misleading way. Finally, there is much detailed knowledge resulting from individual experimental observations that never made its way into a database. Aproppriate ways need to be found to keep this specialized knowledge near the data from the large databases to avoid the danger of the details slipping through our hands.

This consideration raises two questions. First: How can knowledge be passed on reliably if not directly from mouth to ear? Second: How can we find the proper information within all the data we have? Fortunately, there have been examples in the past where people have extremely complex data management tasks successfully: Aeroplanes have been built, although no single person knows all the technical and production details. People know how to maintain and navigate a Boeing 747, although nobody in person has read the whole 1,000,000 pages manual [1] completely. On the other hand, there are also examples, where knowledge got lost over time. It seems unlikely that we still could mason an entire gothic cathedral without steel and concrete. Obviously, direct communication between people played a major role in transferring the proper aeronautic and masonry knowledge.

How about bioinformatics? Has this factor been underestimated in the field of biological databases? Or will it be sufficient to follow a purely heuristic approach on the quest for knowledge: as long as we find interesting data, the databases must be ok? I think, it is required to cultivate a self-critic view of current methods in order to keep the knowledge conservable on a long term.

Accessing present knowledge from biological databases and journal articles is not trivial. In this particular field of knowledge management many issues from computer science like data modeling, controlled vocabularies and usability meet biological data that rarely uses unambiguous terms and is hard to understand without expert knowledge. The challenges faced when conserving and mining biological data are somewhat different from

[1] www.forces.gc.ca/admmat/cosmat/dmgim/cals/background/calswp/libSGMLwho_e.asp

classical 'hypothesis-driven' research. One might argue, that maintaining databases is no research at all. But, it is unquestionable that efficient data management has become necessary for sustainable theoretical and experimental research.

In this work, I will examine what kind of knownledge about protein structures is available in biological databases, and how it can be used most efficiently. The data from a number of databases will be brought together, resulting in Columba, a protein structure data warehouse. Its main purpose is to facilitate the creation of datasets of proteins. Besides a thorough analysis of the data, the applicability of Columba will be demonstrated on a number of sample tasks.

Using this expertise, datasets will be generated for analyzing how well the atoms in 3D-structures are packed. The study will include membrane proteins, organic ligands, coenzymes, binding pockets and reference data. It concentrates on two measures of protein packing: the proximity of atoms to each other (packing density), and the occurrence of large empty spaces between them (cavities). The analysis will resolve contradictions about packing in the literature, and characterize building principles in important protein regions in atomic detail.

Part I

Introduction

2 Basic processes in life can be explored through protein structures

2.1 Proteins are ubiquitous to life

Proteins are found in an enormous variety of forms and functions inside and outside living cells, and in viruses. The most important functions provided by proteins are enzymatic catalysis, transport of other molecules, regulation of gene expression, building stable fibers, transmitting neuronal activity, movement, blood clotting and immune response. Practically, each function of living organisms depends on proteins.

This immense functional heterogeneity of proteins reflects in their biochemical nature: Being linear polymers with more or less the same chemical composition, the overall shapes and properties of proteins are nonetheless very different (see figure 2.1). Determined by the sequence of amino acids - the *primary structure* - the polypeptide chain folds to higher-order elements - the *secondary structure*. Together, these fold into a well-defined three-dimensional arrangement - the *tertiary structure*. Additionally, several of these folded protein chains can assemble to *quarternary structures* or oligomers.

The three-dimensional structure of a protein as it occurs in vivo determines how it interacts with other macromolecules or metabolites. Thus, knowing the structure of a protein is the key to determining its function.

2.2 Protein structure data can be used to address important questions

Researchers interested in protein structures use them to answer many kinds of questions: First, they can be used to derive general principles of how proteins are built and folded in vivo. Second, they give detailed insight to catalytic mechanisms. Third, they provide information how proteins interact with other molecules, such as metabolites, DNA or other proteins. This enables structures of small and large molecules to be docked to protein structures. Fourth, structures can be used to design inhibitors for specific proteins (i.e. structure-based drug design). Fifth, unknown structures of proteins can be modeled using known structural data (structure prediction). Sixth, physical properties of proteins can be calculated in quasi-realistic simulations (molecular dynamics). Finally, pictures or animations of protein structures are very useful in visualizing elementary life processes for communication and teaching purposes.

2.2.1 Many structural analyses require carefully designed datasets

All these analyses require datasets of protein structures and related information. Fortunately, experimental efforts in structure analysis, genome sequencing, DNA microarrays, mass spectrometry, yeast-two-hybrid and others have piled up an impressive array of data

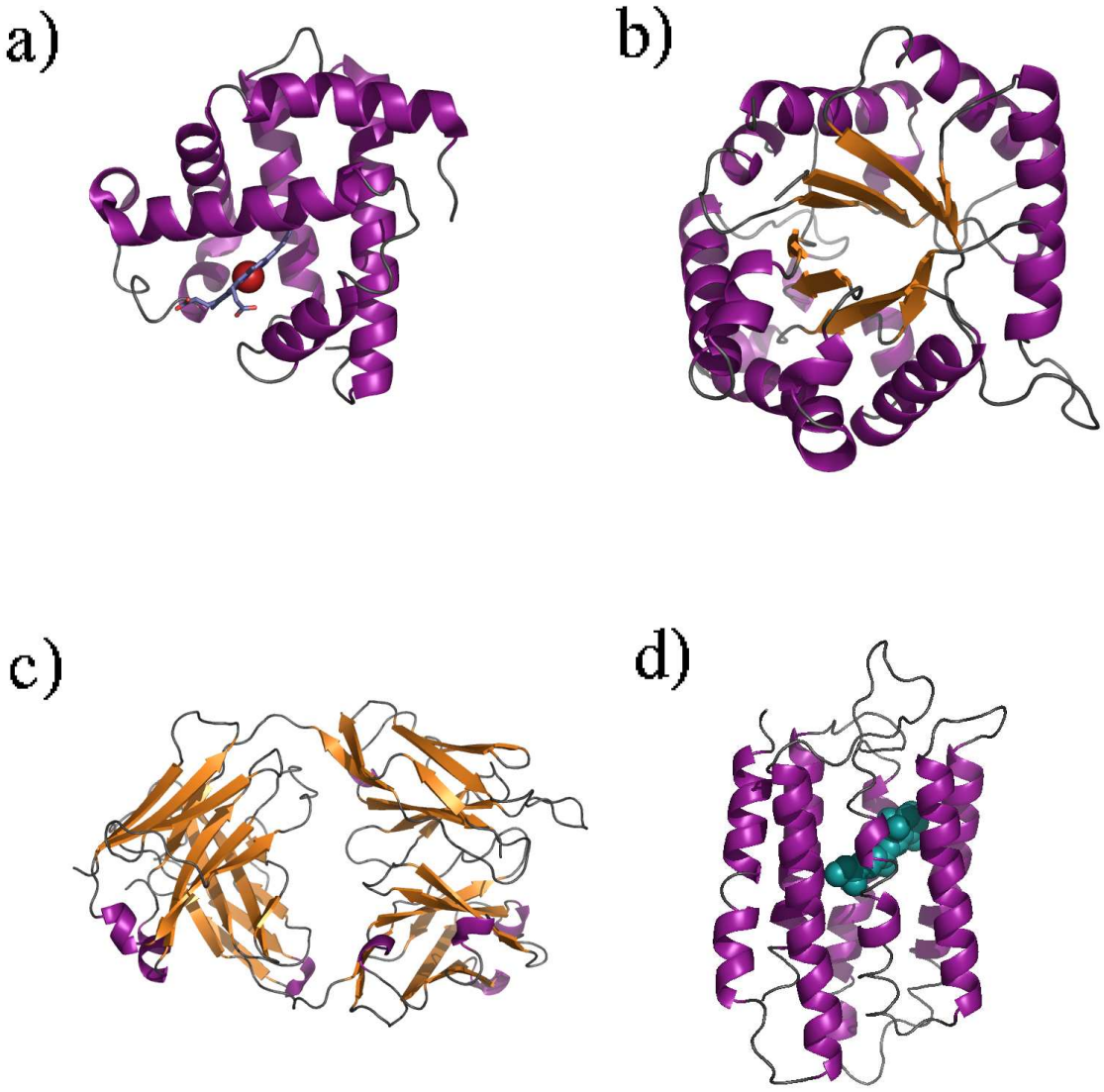

Figure 2.1: Structural heterogeneity of proteins. a) tertiary structure of the oxygen carrier myoglobin (PDB-code 102m), showing a purely alpha-helical fold. b) the glycolytic enzyme triosephosphate isomerase (PDB-code 1aw1), showing the well-known TIM-barrel fold. c) mainly beta-sheet structure of the Fab fragment of a class G immunoglobulin (PDB-code 2jel). d) bacteriorhodopsin (PDB-code 1ap9), a transmembrane protein. The pictures have been created with PyMOL (www.pymol.org).

in the recent years. It is among the most visible achievements of bioinformatics that this enormous influx has been directed to a structured and persistent data storage. But soon after the human genome had been sequenced, it was found that the phrase *'If one piles up enough data, life will be understandable'* does not hold true.

To draw biochemically relevant conclusions and design datasets, the data had to be placed within its biological context.

2.2.2 How can all this data be used efficiently?

Given the data to derive sets of protein structures, one quickly arrives at the main question this work is concerned with: How can all this data be used efficiently? One would expect that the best results are achieved when one can use all the data simultaneously. This is reflected in the fact that the focus of many researchers has shifted from the study of a single gene or protein towards an intra- and inter-species comparison of genes, gene products, metabolic networks and interactions in whole proteomes.

Accessing that much data in short time is not trivial. Both conceptual and technical problems accumulate with the heterogeneity of the databases being accessed. In the context of this work, a comprehensive approach to accessing manifold data about protein structures shall be made. This includes identifying the conceptual challenges, solving the technical problems, and advising a method for creating datasets of structures. As an application, datasets shall be generated, on which the atomic packing of protein structures and possible implications for protein function can be studied.

3 Data on protein structures is organized in databases

Bringing a protein structure to light is an arduous and expensive task. In 2004, it was estimated that a single protein structure determination numbers about 50,000 - 200,000 USD in research costs (David Stuart, personal communication). The two experimental methods solving protein structures on a large scale are X-ray crystallography (Nobel Prize 1962 awarded to Max Perutz and John Kendrew) and nuclear magnetic resonance spectroscopy (Nobel Prize 2002 awarded to Kurt Wuethrich, Koichi Tanaka and John Fenn). In recent years, electron microscopy has emerged as a method to solve structures with a low resolution, complementing the other methods. There is a central repository for all these protein structures, the Protein Data Bank PDB [Berman et al., 2000].

When performing analyses on protein structures, the question asked most often by biologists is: *"Which proteins belong to the same family?"*. Other questions important for reasonably using structural data include *"Do other proteins have the same fold?", "What exactly does this enzyme do?", "Which proteins are associated with a given disease?"*. The data stored in the PDB fails to answer any of these questions directly.

For this reason, numerous secondary databases - resources providing meta-information on PDB entries - have emerged [Carugo and Pongor, 2002]. Many other databases contain links to PDB entries, although they do not primarily address structural data. In the following, both the PDB and multiple types of secondary databases shall be presented.

3.1 The Protein Data Bank PDB

For three decades, proteins with a resolved 3D-structure have been collected by the Protein Data Bank (PDB) [Berman et al., 2000] from the Research Collaboratory for Structural Bioinformatics (RCSB). During that time the PDB has been subject to a number of changes: In the 1970's, only a few structures were known, which were physically stored on punch cards. The PDB data format is still inspired from that era. In the 1980's, it became obligatory to deposit new protein structures in the PDB. In the 1990's the amount of data exceeded the capacity of an optic storage disc, but internet services became common in time to compensate for this. Also, a more detailed and unified data format was enforced. In the early 2000's, the proteomics projects started to produce a great number of structures and the content of the PDB finally exploded (see figure 3.1).

Constant improvements on X-ray methods, the widespread use of synchrotron radiation and the surprising advance of NMR spectroscopy made this development possible. Also, advances have been achieved in solving the structures of membrane proteins [Deisenhofer et al., 1984] and huge multimeric complexes like the proteasome [Löwe et al., 1995] and the ribosome [Ban et al., 2000]. Besides resolving structures of particular large, problematic, and biologically interesting proteins and larger complexes, much effort is being spent on establishing high-throuput methods to produce even higher numbers of structures.

Over the past fifteen years, the average crystallographic resolution of new structures has remained constantly around 2.1Å. A technical advance can be observed by other parameters: the structures have grown considerably in size; the R value, expressing compliance of the structural model with the measurements quality, has dropped by excluding overfitted structures through stricter controls; molecular details, like cis-prolyl residues and unusual conformations in active sites are being resolved more reliably; finally, there is

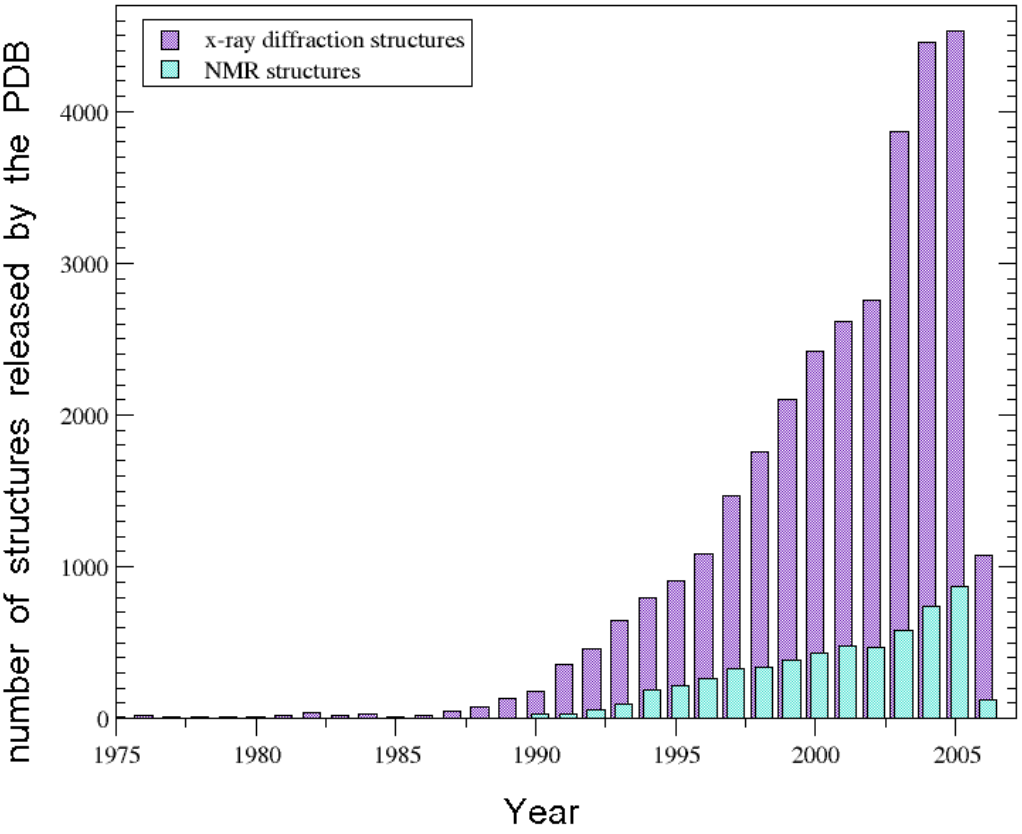

Figure 3.1: Growth of the PDB during the past 31 years. The two most important methods generating protein structures are shown. In 2005 and 2006, many existing structures are missing, because the PDB has not yet released them officially [Berman et al., 2000].

a general trend that more crystallographers also provide the electron density maps along with the structures [Kleywegt and Jones, 2002].

Given the high experimental effort, the number of known protein structures is low compared to the hundreds of thousands of sequences produced by genome projects. The number of more than 35,000 structures in the PDB to analyze in 2006 is still impressive enough to keep busy many protein structure-related researchers.

Each PDB entry contains a header containing a textual description of the molecules in the entry, its authors, literature references, experimental conditions, and a set of 3D-coordinates of atoms. Often links to external databases are contained in addition to the full text information. However, PDB entries are not curated, only archived by the PDB team. This has two consequences. First, the database references are not constantly updated and therefore quickly become out-of-date. Second, the annotation in the PDB header provided by different submitters is highly heterogeneous and not standardized [Bhat et al., 2001]. As a consequence, searching the PDB for annotation is an error-prone task. Annotation may be incomplete or inconsistent with standard nomenclatures. Spelling errors and uncontrolled usage of abbreviations prevent an efficient textual search, and literature references or links to functional and structural databases may be outdated

or missing. This lack of search options has led to a number of second-party databases that parse PDB entries and attach a wealth of links to relevant databases.

Currently, the PDB maintainers have joint forces with groups in the UK and Japan to improve the quality of the data by manual curation, enrich it with additional data, and to build a modern database interface [Berman et al., 2003].

3.2 Fold and family classification databases

Since 1954, when Linus Pauling predicted the structure of alpha-helices, it is known that proteins have common structural features. From 1962 on, when complete proteins could be resolved, it was soon discovered that their overall structures are sometimes very similar, too (e.g. myoglobin and hemoglobin). In the 1980's it was discovered that proteins with homologous sequences are likely to have the same fold [Chothia and Lesk, 1986]. As soon as large numbers of structures were known, the systematic classification of protein folds began. Manual annotation by protein structure experts led to the hierarchical classification in the SCOP [Murzin et al., 1995] and CATH [Orengo et al., 1997] databases, which are analogous in many aspects [Hadley and Jones, 1999]. Automatic methods comparing folds produce structural alignments, many of which have also been deposited in databases like HSSP [Dodge et al., 1998], DALI [Dietmann et al., 2001] and CE [Shindyalov and Bourne, 2001].

Today, according to the SCOP database, about 1,000 distinct folds and 3,000 protein families are known. From the distribution and discovery rate of folds and families, there has been an effort to estimate their total number in nature. These numbers are still a matter of an ongoing debate, with estimates ranging between 400-10,000 different folds and 1,000-30,000 protein families [Liu et al., 2004b].

3.3 Protein sequence databases

The number of sequence database entries is generally much higher than that of protein structures. Sequence databases are well-connected with each other, and they contain much useful functional annotation and database cross-references that are not found in PDB entries. Therefore, this data can be used to improve the annotation of structures for which data in sequence database entries exist.

The most important protein sequence database is UniProt [Bairoch et al., 2005]. It consists of its predecessor Swiss-Prot for which entries are curated manually, and TrEMBL which contains translated sequences from the EMBL database and most known genomes but with considerable redundancy [Boeckmann et al., 2003]. The InterPro [Mulder et al., 2005] database contains a large number of sequence patterns that grouped together thousands of protein families. The SYSTERS database [Krause et al., 2000] calculates protein families from large sequence databases using sequence alignments, but an hierarchical cluster algorithm instead of a fixed sequence identity threshold.

Linking protein sequences from the PDB to the correct sequence database entries is not straightforward. In many protein structures, single residues or loops are missing because they were too flexible to be resolved in the crystallographic electron density map. Also, crystal structures often contain only one or a few domains of a multi-domain protein. Finally, proteins from the PDB can differ from sequence database entries either by point mutations, or because the sequence was altered artificially.

For these reasons, both sequence alignments and manual curation are required to

establish links from PDB structures to sequence databases. The PDBSprotEC database [Martin, 2004] provides a curated list of such database references.

3.4 Databases describing enzymatic and metabolic function of proteins

The functions of proteins are so diverse that it is difficult to categorize them properly. After all, enzymes have been almost completely categorized by four digit enzyme classification (E.C.) numbers. For instance, Triose-phosphate isomerase (TIM) has the E.C. number 5.3.1.1. The first digit indicates that TIM belongs to the enzymatic class of *isomerases*, the second characterizes the bonds that are changed (*intramolecular oxidoreductase* for TIM), and the remaining two digits specify the substrate metabolized by the enzyme. The ENZYME database [Bairoch, 2000] is a catalog containing all E.C. numbers, names and textual descriptions of the enzymatic functions. More details on enzymes are found within the BRENDA database [Schomburg et al., 2004]; in some cases even kinetic data is contained there. The KEGG project [Kanehisa et al., 2004], among other things, groups enzymes to metabolic pathways and also groups other proteins to functional processes, e.g. translation or signalling pathways. Additional functional annotation using a controlled vocabulary and ontology has been made by the Gene Ontology Annotation (GOA) project [Camon et al., 2004].

These databases characterize protein function based on artificial definitions in contrast to structural data which is based on direct experimental evidence. They can be associated with PDB entries either directly by the E.C. numbers or via Swiss-Prot entries. The PDBSprotEC database [Martin, 2004] also offers a carefully curated list of references from PDB chains to E.C. numbers. Finally, the Nucleic Acid Database (NDB) [Berman et al., 2002b] maintains a list of PDB entries of DNA-binding proteins.

3.5 Non-redundant subsets of protein databases

For statistical reasons, scientists are often interested in removing redundancy from their datasets. For that purpose, several databases offer precalculated subsets of PDB chains, in which proteins with a sequence similarity above a given threshold are removed. The first of this kind was PDB_SELECT [Hobohm et al., 1992]. It has been detached by PISCES [Wang and Dunbrack Jr, 2003], which offers not only a high number of precalculated lists with different homology and resolution constraints, but also the opportunity to calculate custom lists with given parameters. Recently, the PDB team has established its own sequence homology clustering method, which is based on the cd-hit algorithm [Li et al., 2001].

3.6 Small molecular compound databases

The PDB has a counterpart in which data of small molecules is deposited, the Cambridge Structural Database (CSD) [Allen, 2002], containing data of more than 300,000 structures. The PDB contains a small fraction of them as protein ligands, which are extremely important as a learning set for structure-based drug design. Therefore, several databases have processed the ligands from the PDB for search purposes. Two of them, LIGBASE [Stuart et al., 2002] and Ligand Depot [Feng et al., 2004] just offer a catalog of the PDB ligands in a conveniently searchable way. The latter employs Marvin, a 2D-structure editor.

Recently, efforts were taken to build specialized databases of low molecular weight

compounds to facilitate design of drug molecules for specific targets: In the SuperLigands database [Michalsky et al., 2005], multiple conformations for all ligands from the PDB are stored. The Relibase also offers various options to retrieve small molecules by chemical similarity [Hendlich et al., 2003]. An analogous project, the SuperDrug database, exists for well-characterized drugs [Goede et al., 2005].

3.7 Other useful resources on protein structures

Crystallographers perform a number of quality checks on protein structures before releasing them to the public. Many of them are combined in the PROCHECK and WHAT_IF [Hooft et al., 1996] program suites. Both applications create reports that indicate untypical parameters of a structure. The Dictionary of Secondary Structures in Proteins (DSSP) [Kabsch and Sander, 1983] is a program that calculates secondary structures from the hydrogen bonds in polypeptide chains. It has been the de facto standard for the detection of secondary structures for a long time. Most secondary structures in the PDB are calculated according to DSSP. Catalytic reaction centres have been collected in the Catalytic Site Atlas (CSA) [Porter et al., 2004].

In addition, there is a number of protein structures that undergo large conformational changes in their normal biological activity. Examples, in which structural data documents such conformational shifts taking place, have been collected in the Database of macromolecular motions [Echols et al., 2003].

Finally it has to be noted that a large number of other databases exists, which are not presented here. The above enumeration contains those structure-related databases that have highly developed user interfaces, are well-known or mark a groundbreaking conceptual progress. A current list, the Molecular Database Collection is being maintained by the NAR journal (*http://www.oxfordjournals.org/nar/database/c/*). This work focuses on a smaller number of databases that can be found in table 7.1 in the Methods section.

4 Data integration gathers biological data by technical means to use it efficiently

"Data integration requires three things: a clearly defined goal, stable source data and a stable data model (N.N., Symposium on Integrative Bioinformatics, Bielefeld 2005)."
Services from all groups of databases can be used intuitively. Their web interfaces are easy to use, but flexibility is limited. When a complex question requires queries on several databases, the frequently occuring hyperlinks allow one to track the desired features across web pages and databases manually. This can be comfortably done for less than ten structures only. For more structures, the ambitious user ends up scavenging the content in a 'clickathon of cut-and-paste, screen-scraping and related disciplines'. For large datasets, this procedure is absolutely unsuitable.

In other cases, users like to perform complex queries that exceed a web servers' capabilities. To overcome both problems, many databases release their data in a structured, machine-readable format. To answer complex questions, the data from two or more databases needs to be accessed from within one computational framework. Making this technically feasible is called data integration.

4.1 Challenges in integration of biological data and possible solutions

Data integration is the more difficult the more different sources of data have to be included. The necessary effort probably grows exponentially rather than linearly with the number of source databases. The reasons for this are of semantical, technical and sociological nature [Stein, 2003].

4.1.1 Semantic challenges

Databases have to simplify. Biological data collections are always based on a simplified model by which they are trying to depict a particular subtopic of biology. The complexity found in nature is almost infinite from a database designers point of view. This has caused the vocabulary, that is used by biologists, to fall into several overlapping sub-vocabularies for particular fields. Often, same names may mean different things depending on the context. Terms like gene, allele or domain are known to conflict when different databases meet. Thus, a database user needs to be aware of exceptions where the model used by the database creators just won't apply.

Among the well-known pitfalls of this kind is that the Swiss-Prot protein sequence database contains only a few sequences of immunoglobulins. Their sequences created by physiological recombination are therefore heavily underrepresented in the database. Other proteins, like the enzyme Aconitase, undergo large conformational shifts or, like the prion protein, refold completely. These properties are not represented in the SCOP fold classification database. When multiple databases are combined, data from such a simplified model can correlate with other data, where it is biologically less meaningful.

The descriptors found in of many biological databases are incomplete or not detailed enough. This imposes severe limitations on the kind of questions a database is able to answer. Because hypothesis-driven research must not be restricted *a priori* by a data model, it follows, that it is necessary to keep data models flexible enough to accomodate further data, which is suitable to test a hypothesis, and not *vice versa*.

Textual fields are often ambiguous. Most database entries contain textual fields that are not standardized. This results in entries that software will not recognize as being similar (e.g. for the species *'Escherichia coli'* in the PDB the following synonymous terms *'E COLI', 'E.COLI', 'E$ COLI', 'ESCHERICHIA COLI', 'ESCHERICHIA C'* and others occur. Even the same words can mean different things (homonyms): General terms like *'domain', 'source organism', 'interaction', 'activation'*, and *'binding'* and protein names like *'src-homologous', 'Rnt related', 'similar to mammal carboxypeptidase'* are known to be problematic. This kind of problem aggravates when several databases are being combined.

To counter inconsistent or overlapping use of scientific terms, many of them have been defined clearly, and compiled manually to *controlled vocabularies*. When semantic relations between terms from a controlled vocabulary are added, this is called an *ontology*, indicating higher-order terms and simple semantic associations. The Gene Ontology (GO) project [Consortium, 2004] immerses both concepts by defining about 16,000 terms related to gene and protein function.

Databases are redundant. Most biological databases contain a high level of redundancy. This is not a primary problem for the creator of an integrated database, because redundancy only increases the volume of data. Given the storage capacity of modern computer systems, it is no technical problem in most cases either. But, finally the user is confronted with the semantical problem of sorting out redundant entries out of datasets he is retrieving from an integrated database, and the difficulty of this certainly rises with the volume and complexity of the data. For these reasons, biologists need to describe their context precisely before using databases.

4.1.2 Sociological challenges

A typical biological database consists of a relational database that contains the data, scripts querying it and building web pages, and a web server. For controlling data quality, in-depth biological expertise is required. Thus, the skills necessary for data integration range from pure biology to pure applied computer science. A single database-maintainer can theoretically master these tasks, but should be networked with more specialized scientists, who can be approached for advice. In a team, coordinative tasks will require an additional effort, depending on team size. This includes not communication between team members, but also subtle team-dynamic effects that can have non-linear effects on the project schedule [Brooks, 1975].

Despite many databases that are increasingly using automatic methods to maintain their data, manual inspection is often necessary due to the complexity of the matter, putting a huge work load on annotators. If they cannot keep pace, missing and false links accumulate, thus lowering the utility of database cross-references and increasing the danger of deriving false conclusions.

4.1.3 Technical challenges

Databases provide their data in a multitude of formats and access methods. Their update cycles differ considerably, and sometimes the underlying data model is changed. While the latter effect, called *database churn*, made large-scale data integration very impractical a few years ago, data models tend to be more stable and consistent today. Many data providers follow a *'code of conduct'* proposed by Stein [Stein, 2002] that lowers technical

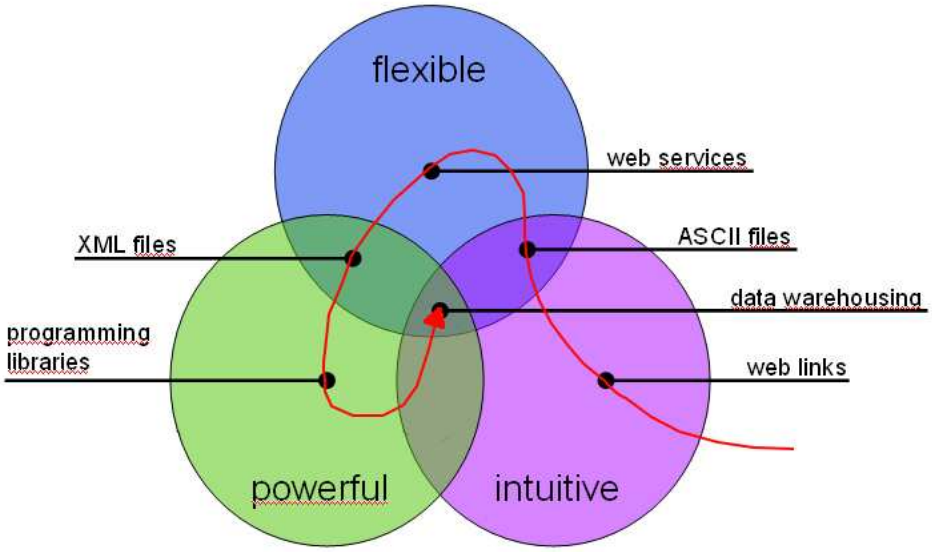

Figure 4.1: Important data integration techniques. The red arrow indicates the progress from human-readable to machine-readable methods.

barriers. It demands that databases provide *globally unique identifiers*, preserve them over time, and offer data in both human-readable and machine readable formats.

A number of data integration techniques are commonly being used. In figure 4.1, their main advantages are displayed, and they are ordered from human-centric to computer-centric methods.

Web links The most convenient method for human users is cross-referencing other databases. It is used by almost all databases in the form of web links. They allow for the quick exploration of the knowledge space around the object of interest. This approach has also been termed *link integration*. The highest interlinked databases such as Swiss-Prot [Boeckmann et al., 2003] or GeneCards [Rebhan et al., 1998] provide links to more than 50 different other databases.

However, cross-references can be easily misleading: A missing cross-reference suggests that no meaningful connection between two particular biological objects exists. However, the absence of a link can also mean that the database providers could not keep up with the growth rates of their own database and the databases they link to. For the same reason, there is a danger that cross-references become obsolete when the linked database changes. Database providers are aware of this problem, and there are approaches to check references automatically to avoid it [Boutselakis et al., 2003, Reichert and Sühnel, 2002].

Plain ASCII files. The main disadvantage of web links is that it is troublesome to process large numbers of them. Providing an entire database as downloadable ASCII text files is a popular relief. Most of these files can be loaded as spreadsheets or parsed easily, and are easy to understand. Despite there is no global standard on their structure, ASCII files have become one of the central hubs for data integration, because entire databases

can be handled without much trouble.

Web services. Users interested in single entries or subsets of a database are likely to use a web service. Both human-usable HTML forms and protocols like SOAP, XML-RPC, .NET and Corba supporting queries from remote programs fall into this category. Compared to the previous methods, web services are more expensive to build and to maintain. They are very useful for retrieving entries such as the dbfetch feature of UniProt does [Bairoch et al., 2005], and for submitting simple queries using the wgetz tool from SRS [Zdobnov et al., 2002b]. The Distributed Annotation System DAS [Dowell et al., 2001] crosslinks web services to integrate heterogenous resources for gene annotation. This combined approach has been termed knuckles-and-nodes [Stein, 2003].

XML files. When databases return sets of complex data, a table will not suffice to represent it. The standardized hierarchical format XML can be used instead. Many XML parsers are available, but the user still needs to infer semantics into the parsed data on his own. Also, it is less convenient to browse through a XML document containing many nodes and sub-nodes than through a table. XML is very useful for representing hierarchical data and networks, such as the metabolic pathways from KEGG [Kanehisa et al., 2004].

Programming libraries. To use data from any of the preceding services with minimal effort, various programming libraries have been developed. They contain parsers, interfaces to web databases and bioinformatical algorithms, thus enabling a user to do almost anything with the data. The price for this is that programming skills that exceed those of most lab biologists are required. The Bio* projects [Stajich et al., 2002, Hamelryck and Manderick, 2003, Mangalam, 2002] are a well-known example for this category.

Data warehousing. Knowing the advantages of these techniques, it seems desirable to unite all of them. A data warehouse comes very close to this: It stores data from many sources in the same place, keeps them connected, and offers convenient methods to access and query it. Thus, a data warehouse combines the simplicity of tables with the ability to represent complex data. It allows to have an intuitively usable web interface built on top, and still can be accessed by software for complex tasks.

Building a data warehouse is a complicated and nontrivial matter, because all of the semantical and technical issues of each constituent database need to be considered. Special attention needs to be payed to update cycles and other changes in the source data. Relational databases have been demonstrated to be able to meet all of these requirements.

4.2 Relational databases are a key technology for data integration

Today, relational databases are the *de facto* standard tool for storing, analyzing and integrating biological data. A **relational database management system (RDBMS)** is a software that maintains relational databases. The RDBMS stores the data physically, protects it against device and power failure and provides interfaces to programming languages, networks, backup systems and other RDBMS'. A relational database is defined by a **data model**. The data model contains schemas, containing tables with precisely defined data types in their columns. For an overview, see figure 4.2. Specific constraints on these entities may be formulated, e.g. excluding empty values or duplicate values, or restricting use to specific users. Other data structures connect tables (foreign keys) and

accelerate queries (indices). Due to the strict constraints imposed by the data model, the value of a database will stand and fall with the aptitude of the data model to the problem adressed.

Using the formal language SQL, a defined set of data can be requested from the RDBMS. In contrast to programming languages, SQL queries are formulated algebraically, not procedurally. As a result, filtering data using SQL queries is less complicated and error-prone than using a programming language. The RDBMS usually optimizes SQL queries towards fast execution. For very complex tasks, SQL queries will not suffice. Programming languages integrated to a RDBMS or closely interacting with it solve this problem. The effort of setting up a RDBMS, designing a data model and filling it with data is high compared to a purely script-based approach. However, the larger a project is, the sooner benefits in maintaining and analyzing the data will put the RDBMS in the advantage.

The main drawback of relational databases is that graphs or other associative data need to be translated into a relational, table-based representation. When querying biological data, such as networks and trees, over several nodes, the query statements will get both slow and clumsy. Possible solutions to this problem have been compiled recently by Trissl [Trissl and Leser, 2005].

4.2.1 Using data warehouses for data integration

As indicated in 4.1.1, defining a data model covering heterogeneous sources poses a major challenge for the design of any data integration system. The approach most commonly described in the literature is schema integration. Schema integration is defined as integrating the schemas of existing data sources into a global schema. This is done by unifying the representation of semantically similar information that is represented in a heterogeneous way across the individual sources [Lakshmanan et al., 1993]. Semantically equivalent attributes or concepts within the different sources need to be identified. The definition of a strategy for merging them into a resulting schema must cover all the aspects of the contributing sources. Recently, a number of tools have been proposed which can aid the user in this task by analyzing names and relationships of schema elements [Do and Rahm, 2002, Rahm and Bernstein, 2001].

4.2.2 Relational schemas for PDB data

The PDB originates from the early 1970's, where punch cards were a state-of-the-art way of storing large amounts of data. Still, PDB files are fixed-width ASCII files. Additional information on the structures is often inconsistent, has many errors or is simply unstructured. This is especially valid for old structures.

From this lack, two projects that aim to provide a clean, structured form of the PDB data in a relational database arose. The first project is the OpenMMS package [Greer et al., 2002] from the RCSB itself. The second aproach is the MSD database [Golovin et al., 2004] at the European Bioinformatics Institute (EBI). Although their scope is limited to the PDB, both can be valuable resources for further integration, because they offer all data from PDB files in a clearly structured form and, partially in high quality. While both the MSD and the OpenMMS data models number more than 100 tables containing everything up to atom coordinates and anisotropic b-factors, the main problem is how to design a query in reasonable time.

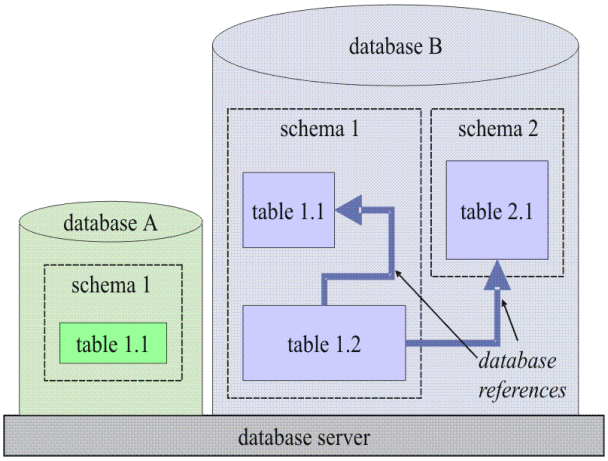

Figure 4.2: Structure of relational database systems: A database server (RDBMS) manages one to several databases (the oil tank-like structures). Each database consists of one to several schemas (dashed frames) which in turn contain none to many tables (boxes). Tables can be interconnected by foreign key references (arrows).

4.3 Existing integrated databases on protein structures

There are several databases in the world wide web that provide information related to protein structures, summarized in table 4.1. They can be divided into three groups:

The first and most frequent approach to the interconnection of data on protein structures spread over multiple original data sources are database cross-references (see 4.1.3). Examples are the IMB Jena Image Library [Reichert and Sühnel, 2002] and PDBsum [Laskowski et al., 2005]. Both store hyperlinks to external databases and not the information itself. Therefore they are well suited for human browsing of single entries, but inadequate for working with sets of structures and their properties.

For a more efficient handling of datasets, the second group has physically integrated data into a single database, or data warehouse (see 4.1.3). In the protein structure world, four such databases can be found: 3DinSight [An et al., 1998] integrates data from PDB, PROSITE, Swiss-Prot, and the Protein Mutation Database. Its focus is on visualization of sequence features, such as PROSITE domains, in the 3D structure. iProClass [Huang et al., 2003] concentrates on protein sequences and contains links to 50 different databases. It can be searched using full-text or sequence similarity search but has no options to search the fields of particular data sources. The PFDB [Shepherd et al., 2002] contains CATH, Swiss-Prot and Gene3D, and also annotates virus domains. Finally, BioMolQuest [Bukhman and Skolnick, 2001] integrates a total of four data sources, i.e., PDB, Swiss-Prot, CATH, and ENZYME.

In 2006, the Protein Data Bank has launched a new web interface to provide not only the links to related sources, but the actual information from SCOP, CATH, and the Gene Ontology and Swiss-Prot references. The same information is also available from the curated PDB of the Macomolecular Structure Database MSD [Velankar et al., 2005].

The third group contains the sequence retrieval system (SRS) [Zdobnov et al., 2002a] that focuses on building complicated queries over multiple databases. However, SRS is a general purpose data integration system and lacks the specific protein structure orientation of the other databases, i.e. SCOP and CATH are not included in SRS. The

Site	integrated databases	access methods	comments
Jena Image Library	7	HL, FLD, ASC	-
PDBSum	15	HL	-
BioMolQuest	4	HL, FTS, FLD	-
3DinSight	4	HL, FTS	focus on protein domains
iProClass	∼90	HL, FTS	focus on sequence data
SRS	∼200	HL, FTS, FLD, ALG	focus on sequence data
MSD	10	HL, FTS, FLD, ALG, ASC	curated PDB data
New PDB site	5	HL, FTS, FLD, ALG, XML	-

Table 4.1: Existing integrated databases containing information on PDB structures. For each database, the number of integrated sources and possible access methods are listed. The abbreviations for available access methods are: **HL**-hyperlinks, **FTS**-full text search, **FLD**-field-specific search, **ALG**-algebraic combination of queries, **ASC**-ascii output, **XML**-xml output.

query interface of SRS is very advanced and allows very subtle combinations of database requests.

All these integrated databases have in common that they either cover only a small range of the protein structure-relevant sources, or that the query capabilities are too limited to allow efficient cross-database queries. In short, there is no integrated database yet that is both flexible and rich in its content.

5 Application: Packing analysis of protein structure datasets gives clues about their quality and function

Most protein chains fold into well-defined structural domains. In this respect, the linear polypeptide is less similar to a wound-up pearl chain, whose monomers (the pearls) can rearrange easily without disturbing the other pearls, than to a twisted steel chain, at which the members are not freely rotatable against each other. Thus, the chain will easily get locked in certain arrangements. Protein chains, constrained by a set of interaction forces like hydrogen bonds, salt bridges, nonpolar interactions, disulfide bridges and sterical constraints show a similar behavior. A stable protein structure should satisfy these constraints as good as possible. Taking into account that amino acids have very different sizes, it seems difficult that a protein chain can fold without enclosing empty spaces in its interior.

How much space is there between atoms in the tertiary structure? Are the atoms arranged in an almost optimally dense packing of spheres, or does the protein interior contain cavities? In this study, methods of packing analysis shall be used to answer these questions for tailored datasets, in particular membrane proteins, ligand- and coenzyme-binding sites, and a reference set. There are two main approaches by which protein packing can be described: packing densities and internal cavities.

5.1 Packing density of protein atoms

In the literature, there are contradictions about how the deepest portions of protein structures are packed. Tsai compared the packing of deeply buried atoms to surface atoms and found that the packing density is highest in the deep interior and that few or no cavities occur there [Tsai et al., 1999]. In an earlier report Hubbard found atom-sized cavities most frequently in the protein core [Hubbard et al., 1994], which is contradictory to the above statement.

The packing density quantifies how much space there is around the Van-der-Waals sphere of an atom in relation to the space inside it. To measure packing densities, the space inside a protein structure is partitioned among individual atoms, and atomic volumes are calculated for them. Hydrogen atoms are usually represented implicitly using slightly increased Van-der-Waals radii (also termed atom radii). With the atomic volume inside the Van-der-Waals radius V_{VdW} and the volume beyond it V_{SE}, the packing density of an atom PD is defined [Gellatly and Finney, 1982] as

$$PD = \frac{V_{VdW}}{V_{VdW} + V_{SE}}. \tag{5.1}$$

The problem of dividing space into polyedric bodies around fixed points was originally solved by the Voronoi procedure [Voronoi, 1908]. It has been further developed to include distinct atomic radii such as the Richards B Method [Richards, 1974] and the Radical Plane Method [Gellatly and Finney, 1982]. Additional improvements have been made, including the use of curved instead of planar interfaces between atoms for a more reasonable allocation [Gerstein et al., 1995, Goede et al., 1997] (see figure 5.1). On the protein surface, the atomic volume is limited by the center of a water-sized probe rolling over the Van-der-Waals spheres of all atoms [Gerstein et al., 1995]. An extensive analysis of selection criteria for structures and atom sets has been done, which focuses on statistical issues of Voronoi-based methods [Tsai et al., 2001].

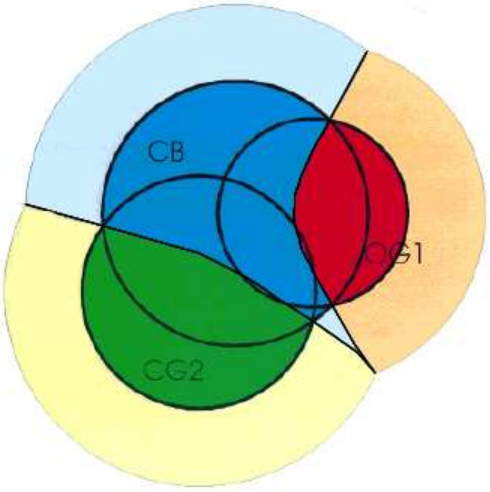

Figure 5.1: Improved Voronoi procedure with hyperboloid faces [Goede et al., 1997] applied to the side chain of threonine. The dark colored circles represent the Van-der-Waals spheres of the three side chain atoms. The light colors show a layer of 1.4Å around the VdW sphere, the solvent excluded volume. The colors indicate to which atom the space has been assigned (blue:β-carbon, green:γ-carbon, red: hydroxyl group). For a diagram describing how the partition is constructed, see figure 8.2.

The alpha-shape method [Liang et al., 1998a] and the occluded surface method [Pattabiraman et al., 1995] provide alternative means to calculate quantities describing packing of protein atoms. These have been demonstrated to procude results comparable to those of the Voronoi-based methods.

The protein interior is packed at least as efficiently as small organic crystals [Richards, 1974, Chothia, 1975]. Aliphatic groups are packed more efficiently than peptide bonds and charged groups [Harpaz et al., 1994]. The packing density was found to depend on protein size, secondary structure and amino acid composition, but not on crystal temperature; it is similar in homologous protein structures even for distantly related proteins [Fleming and Richards, 2000]. By comparing the packing of the protein interior to model liquids and solids, it was determined that proteins more closely resemble randomly packed spheres than jigsaw puzzles or organic crystals [Liang and Dill, 2001]. It remained unclear, however, how packing in the protein interior differs from the surface to the deepest parts of a structure.

5.2 Internal cavities in protein structures

Some protein structures contain locations, where a water-sized probe can be placed in the interior such that it neither intersects any atoms' Van-der-Waals sphere nor reaches the surface without such an intersection [Richmond, 1984]. These locations are defined as cavities.

Empty space inside tightly packed structures would be thermodynamically unfavorable. The given free energy for creating a polar cavity of 1.4Å radius is about 21 kJ/mol [Kocher et al., 1996], which is in the range of the stabilization energy of a whole protein

(20-60 kJ/mol). In a more recent study, the destabilisation energy for the removal of a methylene group in the hydrophobic interior was found to be 5 kJ/mol [Loladze et al., 2002]. In crystal structures of proteins, many cavities are occupied by water molecules, mitigating the destabilizing effect. Other cavities appear empty, because many water molecules are not detectable in X-ray structures due to delocalization. Most NMR structures contain no water molecules at all.

5.3 Sites where packing has functional consequences

Various mutation experiments have proved that filling a cavity increases thermal stability [Ishikawa et al., 1993] and, vice versa, introducing a new cavity decreases it [Matsumura et al., 1988, Sandberg and Terwilliger, 1989, Eriksson et al., 1992]. Nevertheless, thermophilic and mesophilic proteins do not essentially differ in packing [Karshikoff and Ladenstein, 1998]. The necessary stability at high temperatures must be reached by other means. Filling cavities can also inhibit motion of functionally important regions of a protein, thereby diminishing its catalytic activity [Ogata et al., 1996]. This means that tight packing lowers protein flexibility.

There may be a compromise between protein stability and flexibility, resulting in the occurrence of cavities. This hypothesis is supported by an analysis done by Liao [Liao et al., 2005]. In this case, a correlation between the packing density of C_α atoms, hydrophobicity and sequence entropy of residues in several protein families was found. The more variable a particular region of a protein is, the less well it should be packed. A possible application for protein modelling was demonstrated on T4 lysozyme and cytochrome: The fate of cavity creating and collapsing mutations can be reliably predicted by energy minimization [Machicado et al., 2002].

Packing has been analyzed in a number of specific groups of structures, including protein-protein interfaces [Halperin et al., 2004], thermophilic and mesophilic proteins [Szilagyi and Zavodszky, 2000], ribonucleic acids [Voss and Gerstein, 2005], protein-DNA interfaces [Nadassy et al., 2001] and several families, which have been studied in detail [Fleming and Richards, 2000].

There has been no fully convincing analysis of packing density in membrane domains yet. Eilers and colleagues applied the occluded surface method [Pattabiraman et al., 1995] to assess the packing densities of transmembrane helices in 11 helical membrane protein structures [Eilers et al., 2002]. They concluded, that helical membrane proteins are generally packed more densely than other proteins. This conclusion is surprising, because the hydrophobic effect, a driving force for helix-helix interaction, is absent inside the lipid bilayer [MacKenzie and Engelman, 1998, White and Wimley, 1999]. In addition, polar- or hydrogen-bonded interactions generally occur less frequently in transmembrane domains than in water-soluble globular proteins [DeGrado et al., 2003]. The inclusion of prosthetic groups and the pervasion of ion channels and solute transporters with water-filled pores could result in deviations of the molecular packing. Most of these membrane proteins open through gating mechanisms that require broad molecular rearrangements of their transmembrane domains [Locher et al., 2003, Perozo et al., 2002, Swartz, 2004].

The dense packing of secondary structures to each other plays a central role in the folding and stability of proteins [Chothia et al., 1981, Popot and Engelman, 2000, Preissner et al., 1998]. In many membrane proteins, mobility of the transmembrane domain is required for the proper functionality [Jiang et al., 2002]. Given the objections above, it

needs to be analyzed, whether the dense packing of transmembrane helices is contradictory to the necessary mobility.

The known structures of enzymes bind small molecular substrates according to the induced fit model [Gutteridge and Thornton, 2005]. The side chains of residues around the catalytic site rearrange to bind the substrate more tightly. It is reasonable to expect that these tight interactions between protein and ligand are also reflected in local packing. The sizes and shapes of some ligand-binding-pockets have been analyzed [Liang et al., 1998b]. This analysis contained only sparse results, which could explain how the ligands themselves are packed. Similar observations for coenzymes and prosthetic groups are missing entirely.

Protein packing properties have been applied to calculate the intrinsic compressibility of a protein [Paci and Marchi, 1996, Harpaz et al., 1994]. Other practical applications include calculation of packing densities for ligand binding prediction [Kuhn et al., 1992, Liang et al., 1998b], quality assessment of protein structures [Pontius et al., 1996], calculation of partial specific volumes [Tsai et al., 1999], and design of novel proteins [Dahiyat and Mayo, 1997, Kono et al., 1998].

As mentioned above, there is no consensus in the literature about where in the protein interior cavities preferably occur. Other questions, regarding packing in the deepest regions of the protein and packing of membrane helices, ligands and coenzymes, remain unanswered.

6 Tasks addressed in this work

Based on these considerations, two scientific goals for this work are formulated: First, to integrate and analyze information about protein structures in an extent that has not been reached before. Second, to analyze the packing in sets of protein structures created from this data. In detail, the following tasks are to be executed within the scope of this work:

1. Build an integrated database for accessing and analyzing protein structure data.
2. Characterize the data, information and knowledge on proteins found in the database.
3. Make the database available to the scientific community.
4. Construct sample datasets as test cases for the database and devise general rules for creating datasets of protein structures.
5. Create datasets for functionally relevant sites in protein structures.
6. What does the packing of these sites tell about their function?

Part II

Material and Methods

7 Annotation on protein structures from 16 databases is integrated in the Columba data warehouse

To query data on protein structures efficiently, information about them was assembled from many databases. The resulting collection of protein structure annotation was stored in a data warehouse. The aim was to design a database that contains more useful annotation on protein structures than other projects, while being both maintainable and easy to use. Integrating all available data from all available databases would be neither technically nor scientifically reasonable because of the immense complexity of such a database.

Therefore, only the most relevant, well-known and qualitatively reliable data was used. Sixteen databases were carefully selected for integration into the Columba database: Structures from the PDB [Berman et al., 2000] were annotated by fold classification from SCOP [Murzin et al., 1995] and CATH [Orengo et al., 1997]; enzymatic functions from ENZMYE [Bairoch, 2000], KEGG [Kanehisa et al., 2004] and the Boehringer maps [Michal, 1993]; sequence data from the Swiss-Prot [Boeckmann et al., 2003] using links from PDBSprotEC [Martin, 2004]; functional annotation from GO [Ashburner et al., 2000] and GOA [Camon et al., 2004]; taxonomic information from the NCBI taxonomy [Wheeler et al., 2000]; sequence homology families from PISCES [Wang and Dunbrack Jr, 2003]; the sequence clustering done by the PDB itself [Li et al., 2001]; and secondary structures calculated with DSSP [Kabsch and Sander, 1983]. In addition, the Protein Topology Graph Library [May et al., 2004] and the SYSTERS protein family database [Krause et al., 2000] were added for in-house usage.

Each of these databases was defined as a *data source* within the Columba database. The data sources are the central element of Columba's architecture. Each data source can be maintained, loaded and updated independently, and the database can be extended easily by the addition of data sources without changing the existing ones. Each data source consists of a *data model* and a *data workflow* part.

The *data model* defines what data is contained in the database and how it is structured. The data model defines database schemas, tables, columns and how items of different origin relate to each other. This part of the implementation determines how efficiently queries will be carried out. For Columba, a star-shaped data model has been implemented on the open-source RDBMS PostGreSQL 7.4 (see figure 7.1).

The *data workflow* is a program that writes data into the database. It has to take into account all eventualities and constraints arising from both the data and the data model. The data workflow was implemented in the Python programming language.

The database schemas, a metadata table, and the main routines of the data workflow

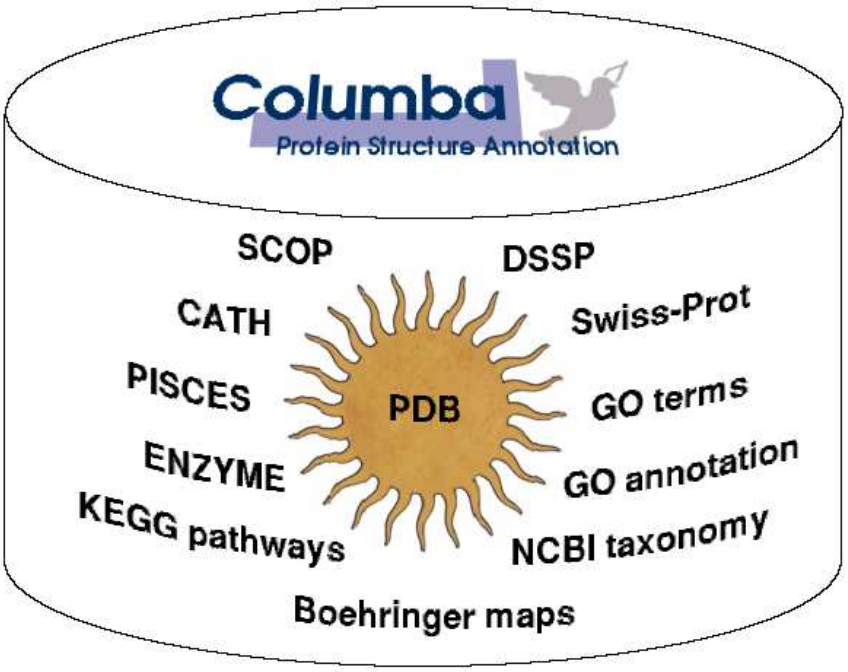

Figure 7.1: The data model of the Columba database is called a *star schema* [Rother et al., 2004]. The PDB is included in the center of the model. All data items from the other integrated databases are directly or indirectly linked to PDB entries.

are not assigned to particular data sources. To demonstrate the applicability of Columba to biological questions, the data sources will be described first.

7.1 Data sources integrated in Columba

In Columba, a data source consists of data, some tables within the data model fit to the data, and a software module that writes the data into the tables. Each data source has a separate set of tables, and each table is assigned to exactly one data source. Thus, data from different sources are never mixed together. The philosophy behind the program is that the responsibilities between tables and modules are unambiguous and non-overlapping. This is very important in following the origin of data items, and improving the maintainability of the database.

For each of the databases included in Columba, a single data source module was created (with the exception of biosql, where two databases are adressed by a single parser). Additional data sources were created to interconnect the primary data. For a complete list of the data sources included in Columba, see table 7.1. Detailed descriptions of each data source can be found in the appendix A.

7.1.1 Additional databases analyzed in this study

A number of additional secondary databases were considered during the project. However, none of them was included in Columba. Some were rejected because the data was judged not informative enough about PDB structures. Others were considered interesting but assigned a lower priority.

Three sets of results from automatic structural alignment procedures were found on

the web: HSSP [Dodge et al., 1998], CE [Shindyalov and Bourne, 2001], FSSP/DALI [Dietmann et al., 2001]. All three were available as easily downloadable text files and reference PDB chains directly. The tabular data available for CE and DALI was about nine months older than the data available through the projects' dynamic websites.

Two integrated sequence databases were also queried: InterPro [Mulder et al., 2005] and UniProt [Bairoch et al., 2005]. The latter integrates several sequence databases as a whole while the former concentrates on protein sequences matched by certain patterns (the InterPro domains). However, both contain database cross-references to the PDB only in the entries from the Swiss-Prot database.

The BRENDA database [Schomburg et al., 2004] contains detailed descriptions of enzyme functions along with specific data for particular organisms and tissues. However, the database is proprietary and was not available in an easy-to-use format. The necessary data, mostly references from E.C. numbers to PDB entries was retrieved using a python script that accessed the database at slow, random intervals in order to keep from violating the conditions for academic use (this method is also termed the infamous *screen scraping* technique). A database of transcription factors, TRANSFAC [Fogel et al., 2005] was considered. Here, a commercial version with more data exists, that was not available. The free version 6.0 of the database contained only 35 references to PDB entries and was not used in this study.

These databases were included in order to analyze to what extent the PDB is annotated by them. The annotation from them was not processed further.

7.2 Constructing a star shaped data model around PDB entries

Columba is centered around PDB entries [Berman et al., 2000]. Descriptions of all protein structures from the PDB are at the very heart of the data model, but the structural data itself is not contained. The other thirteen data sources contain the actual annotation. Each of them is linked to the PDB entries directly or indirectly. Owing to the radial arrangement of the data sources, this type of data model was called a 'star schema' [Rother et al., 2004] (see figure 7.1). To store data from the 16 data sources in a clear structure, the database was split into sections for raw data and sections for queries by end users. These sections are termed database schemas.

7.2.1 Subdividing the database into schemas

The Columba database consists of seven schemas, containing a total of 108 tables (see table 7.2). The philosophy behind this allocation is *'One software - One schema'*. Thus, there is no doubt which program module created a particular cluster of data. Here, the paradigm of schema integration is adopted: The data is primarily loaded into four of the schemas (*biosql, goa, ncbi* and *ptgl*) by external parsers, and useful representations of this data are generated by schema mapping into a fifth, the *columba* schema. Being the most important part of the database, the *columba* schema contains representations of all data sources, making it sufficient to answer most queries. Two schemas, *public* and *web* provide the functionality of the web interface (see A.3.20). An entity-relationship model of the *columba* schema, displaying the tables and the links between them, is given in figure 7.2.

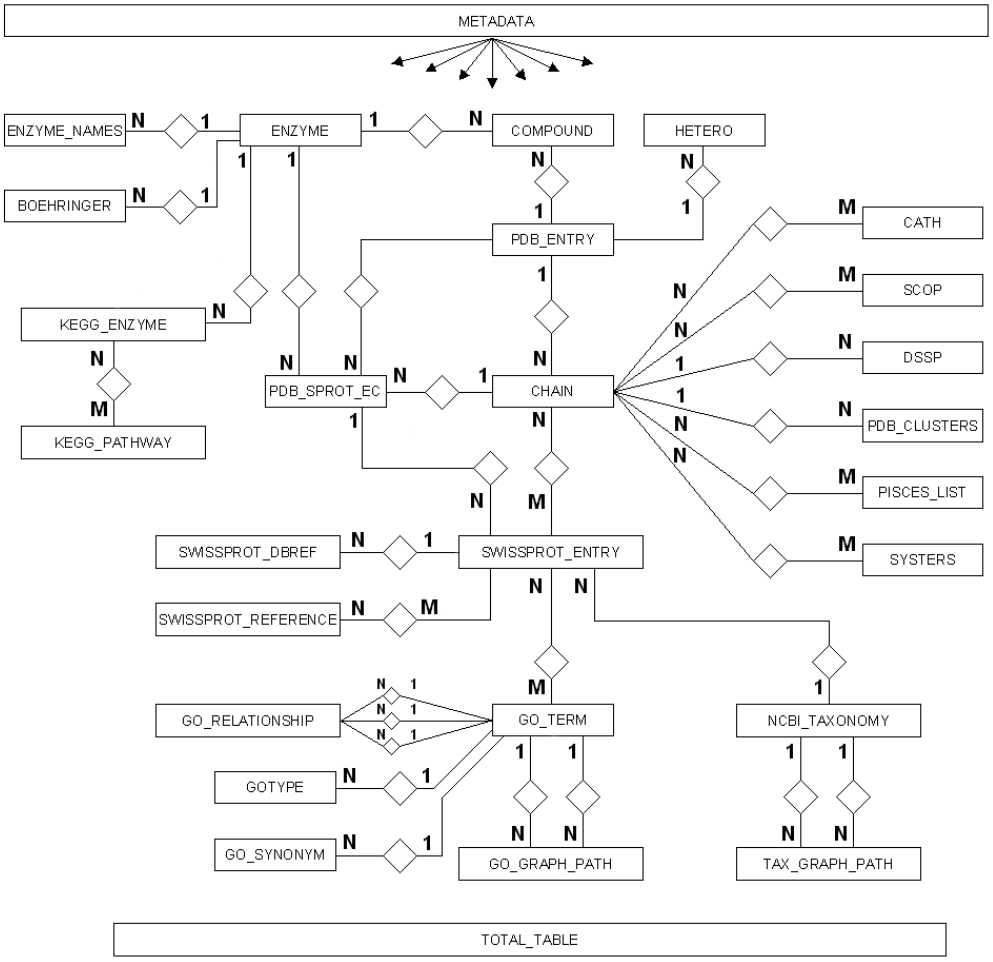

Figure 7.2: Entity-relationship model of the *columba* schema of the database. Each box corresponds to a table; each line to a reference between two tables. Letters indicate the kind of reference: In the *1..N* type, each entry in the first table is referenced by zero to many in the second, but not vice versa. In the *M..N* type, these references occur in both directions. The four tables *pdb_entry*, *compound*, *chain* and *hetero* originate from the Protein Data Bank (PDB). All others, except the *metadata* table, are created by the other data sources. For clarity, not all relations have been drawn.

7.2.2 Representation of PDB entries in Columba

Each entry in the PDB corresponds to a protein structure, and the terms *PDB entry* and *protein structure* will be used synonymously throughout this study. A PDB entry is organized in *compounds* representing biological units, and each compound consists of one or more polymer *chains*. Additionally, PDB structures contain non-polymeric molecules, so-called *hetero groups*. The columba schema contains four tables representing these entities:

Pdb_entry: Each entry contains general information on a PDB structure like its name, an experimental method, resolution, a date on which the structure was deposited, and its authors.

Compound: A compound contains the name of a biologically active unit, the enzyme classification number (if it has one), and information on the source organism or tissue, as given in the PDB header.

Chain: An entry contains the one-character identifier from the PDB files, the number of atoms and residues, and the primary structure in one-letter-code, derived from the atom coordinates in the PDB. As some chains in the PDB are nucleic acids, they have been indicated by brackets in the sequence.

Hetero: Hetero group entries contain the three-letter acronym and residue number of the particular group, the number of atoms and the name of the molecule. Water molecules were summarized to one hetero entry per structure to keep the table more concise.

The first column in each of these tables contains an integer primary key which is generated automatically using the SERIAL data type. Thus, it is ensured that each data item has an unambiguous identifier that can be used for referencing data from that particular table. There are lots of PDB features intentionally ignored in this data model (e.g. atom coordinates, Crystal cell and refinement parameters). Cramming all available data fields into a relational schema would leave the user unconscious of the strengths of the system and would slow down maintenance severely. A relational schema like OpenMMS is more suited for detailed analyses on particular sub-fields of the complete PDB. A description of the tables used by the other data sources can be found in the appendix A.

7.2.3 Metadata

In the *columba* schema, a single metadata table contains entries for all data sources in a standardized format. Here, the name, version and parameters of the program that created particular data are stored. In addition, the date/time at which the program was started are stored, along with the name of the user that started it and manually entered remarks. In each data source, at least one table contains references to a corresponding entry in the metadata table.

7.2.4 Schema mappings

For two of the data sources integrated in Columba (Swiss-Prot/Gene Ontology (*biosql*) A.2.2 and the NCBI taxonomy (*ncbi*) A.2.9), external software was available that wrote the data into a database. Both use the data model of the BioSQL project [Stajich et al., 2002], containing 31 tables. The BioSQL data model is too detailed for the kind of queries

straightforward to Columba. The Gene Ontology Annotation (*goa*) was directly read into a separate schema of the database.

Thus, the data needed to be converted into a more concise representation. First, the original data was parsed into separate schemas (*biosql, ncbi and goa*. Second, a set of rules was written in SQL that translated data from a set of tables in the original schema to a set of tables in the *columba* schema. The process is illustrated in figure 7.3.

Three schema mappings were implemented in the Columba database:

1. ***biosql* schema to *columba* schema:** From 31 tables in the *biosql* schema, 8 tables in the *columba* schema are created, containing protein sequences, GO terms and their relationships. Links to PDB entries are created during the mapping process in two separate tables, using information from the PDBSprotEC database. Also, links between GO terms on different levels of the GO hierarchy were calculated to allow queries of the type 'look for X in the GO terms *and their ancestors*'.

2. ***ncbi* schema to *columba* schema:** Three tables from the *ncbi* schema are combined into two in the *columba* schema. They contain taxa that annotate Swiss-Prot entries and their taxonomic relationships. Similar to GO terms, links between different levels of the taxonomic hierarchy are calculated, allowing queries to be made. The taxa are linked to PDB entries via corresponding Swiss-Prot entries.

3. ***goa* schema to *columba* schema:** From a single table in the *goa* schema, all data is transferred to another table in the *columba* schema. Also, links to Swiss-Prot entries, PDB entries and GO terms are inferred from *biosql_mapping*.

7.3 Modular annotation workflow filling the database

The data from the original sources shows a great variety of formats, such as flat files, database dump files, XML and pure HTML pages. A number of parsers, written in Python, Perl, AWK and C, was used to handle the data. About half of those included were from publicly available projects such as BioPython [Hamelryck and Manderick, 2003, Mangalam, 2002], the others were written entirely from the ground up. To load the Columba database with this data, a Python application was written. It integrates data from the sixteen data sources shown in table 7.1 and writes it into the database using the parsers mentioned above and a number of SQL scripts.

The data workflow is, much like the database itself, organized in a star-shaped manner. A central control program contains one software module for each data source. Each module implements a fixed Python interface and is sufficient to fill the tables that it is responsible for. Thus, any new data source can be included into the data workflow by implementing that interface.

The data workflow implements three use cases:

1. Create the database - Deletes and re-creates the database and creates the schemas, functions and other important settings. No tables are created yet.

2. Update data sources - Creates the tables, parses data and writes it to the database for one or many data sources.

3. Automatic on-demand update - The application looks in the internet for new data files automatically and determines, which data sources need to be updated.

7.3.1 How data sources are updated

Updating data sources follows a strictly defined protocol because duplicate, outdated and skipped entries need to be avoided at all costs. To ensure this, all data source modules implement the same interface, despite the heterogeneity of the data. The general procedure for updates is to completely throw the old data away and fill the tables all over again, even when only minor changes have occurred. This way, a large number of hardly traceable errors can be avoided.

Whenever a data source is updated, (a) first all data for this data source is wiped out. The tables belonging to it are dropped and then created again. (b) Then, the data source module will create an entry in the *metadata* table. At this point, the database is ready for the annotation to be written. (c) Next, the data source submits initial data that does not depend on the PDB data to the database. (d) After initialization, a list of PDB-ID's is looped through. The data source module writes annotation for each PDB entry at a moment to the database. (e) Finally, the data source submits data that requires all PDB entries to be already processed.

When multiple data sources are to be updated, step (a) will first be processed for all data sources, then (b) and so on. Each data source is applied to each PDB entry only one time. The steps (a) and (b) are obligatory for all data sources. The distinction between

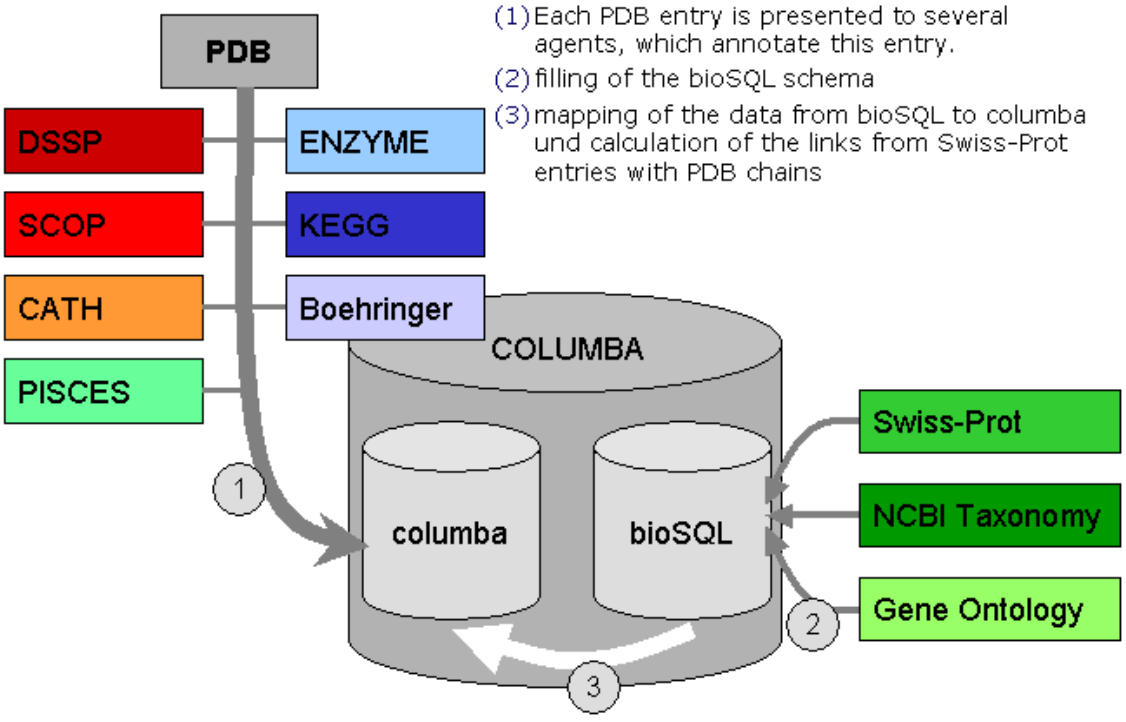

Figure 7.3: Schema mappings in the Columba database. The diagram shows the order in which data is processed. (1) PDB entries are annotated by some data sources directly in the columba schema (not all shown). No mapping is done for them. (2) Data sources that have an own data model along with the parser are loaded to a separate schema. The three modules are all from the BioPerl/BioSQL software. (3) The data is mapped to other tables in the columba schema, and links to the PDB entries are generated.

initializing, handling PDB entries and finalizing, is important in allowing all data to be created in a single program run. Of steps (c),(d) and (e), some are skipped by particular data sources.

7.3.2 Resolving dependencies between data sources.

All database references are unidirectional, meaning that the referenced data must exist in the database before adding data to the referencing table. This imposes logical dependencies between the data sources, which are displayed in figure 7.4 and are also specified in the modules. When updating several data sources, they will be ordered, such that the data sources others depend on will be handled first. Because updating a data source will break all database references from data depending on this data source, the dependencies can and should be used to schedule data sources for updates.

7.3.3 Automatic on-demand update

The data workflow contains a download manager that keeps a list of internet URL's assigned to particular data sources. When performing an on-demand update, the download

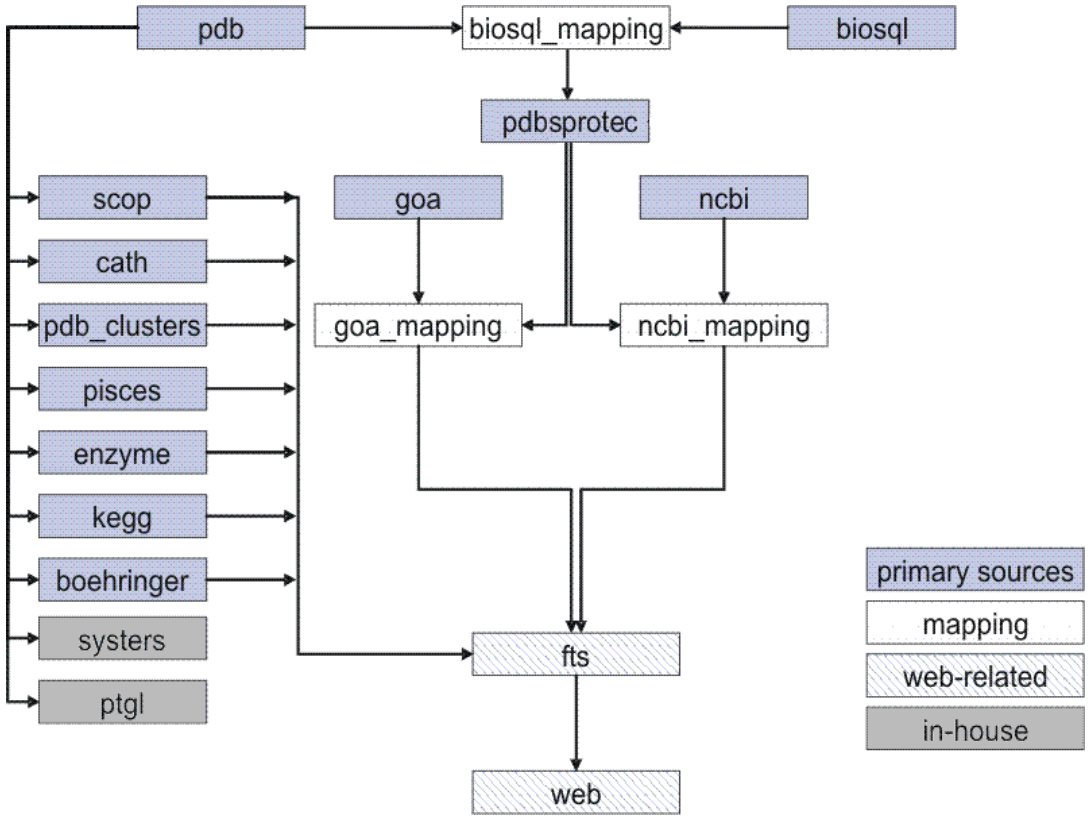

Figure 7.4: Dependencies between the data source modules in Columba that determine the order in which data sources must be processed. The primary data sources (blue) load data for public use, the mapping data sources (white) re-arrange data within the database, the web-related data sources (shaded) are required by the Columba web interface, and the in-house data sources (light blue) load data for internal usage. Arrows point from a required data source to the requiring one (e.g. the PDB must be loaded before any of SCOP, CATH, etc can be processed).

manager checks, whether any of these URL's provides modified or new data files. Such files are downloaded, and the according data source and all its dependant data sources are scheduled for an update. Of the sixteen data sources, 10 can be updated by the download manager; two (pdb, ncbi) have their own procedures to download files; and four (boehringer, dssp, ptgl and systers) are entirely based on local files.

7.3.4 Adding new data sources

The software maintaining Columba has been designed to be highly modular and, therefore, easily extendable. Adding a new data source module will not affect the existing parts of Columba and vice versa. To add a new data source, three steps are necessary: First, the new module needs an unique identifier. Second, a SQL script needs to be written that creates the necessary tables, views and functions in the database. Using the data source ID in their name, ambiguities within the database can be avoided easily. Third, a data source module needs to be written that retrieves and parses the data and writes it to tables. A template code facilitating this task is provided with the Columba source code.

7.4 Analyzing completeness and redundancy of data in Columba

7.4.1 Coverage of the PDB by secondary databases

To find out whether the effect of missing links described in 4.1.2 has a significant impact on protein structure annotation, it was investigating to what extent PDB entries are covered by second-party databases. The linking of all PDB entries to six fold/family classification databases (CATH, SCOP, DALI, HSSP and CE), to four sequence databases (Swiss-Prot, UniProt, InterPro and PDBSprotEC), to four metabolic pathway and enzyme catalogs (KEGG, ENZYME, PDBSprotEC and BRENDA), and to functional annotation from the Gene Ontology Annotation (GOA) was analyzed.

For each data source, the set of referenced PDB entries and PDB chains were compiled, listing PDB-ID/chain-ID, resolution, method and date of structure deposition. These sets were compared to a full PDB set and to each other. The intention was to measure whether two or more data sources cover mostly the same structures.

In order to estimate the overlap of two datasets X and Y, the pairwise correlation was calculated as the amount of common entries, the intersection $X \cap Y$, divided by the number of entries contained in the union $X \cup Y$:

$$overlap := \frac{N(X \cap Y)}{N(X \cup Y)} \quad , \tag{7.1}$$

where $N(X)$ denotes the cardinality of a set X.

We calculated the scores for all pairwise combinations of data sources and constructed a tree to reflect the similarity among the structure sets. The tree was constructed using a modified UPGMA method [Sneath and Sokal, 1973], replacing the composite distances for higher-order nodes with the overlap score for their particular combination of data sources.

7.4.2 Information theoretical measures

This way, the contribution of each data source to the Columba database can be quantified. But are the entries from each data source unique in their way, or does the annotation

correlate between proteins or data sources? To elucidate this, statistical methods need to be applied to the fields within the database. Several measures commonly used for such a purpose originate from information theory.

Since its founding by C. E. Shannon in 1948, information theory has had great impact on communication technology, cryptography and, most recently, bioinformatics. Its main statement is that deterministic and probabilistic knowledge are qualitatively the same. It provides a number of ways to quantitatively analyze information in discrete variables, as well as correlations between them.

For analyzing data in textual fields in Columba, 22 properties were defined each corresponding to a table column linked to PDB chains. Each property was treated as a discrete variable, resulting in one value per PDB chain. These properties were analyzed using the Shannon entropy and maximum redundancy.

7.4.2.1 Shannon entropy

The Shannon entropy H is a measure of the statistical variety within a discrete random variable X. More intuitively, the Shannon entropy is the number of binary questions required to find out a value of X. It depends on the probabilities, at which the individual states $X_1, X_2, ...X_n$ of X occur. The entropy of X is defined as

$$H(X) = -\sum_{i}^{N} p(X_i) * log_2 p(X_i) \tag{7.2}$$

It is easy to prove that the $H(X)$ becomes maximum for $p(X_i) = \frac{1}{N}$, with $H_{max}(X) = -log\frac{1}{N}$. The Shannon entropy H is measured in Shannon $[Sh]$, which are also called bits. This unit is somewhat similar to the bits used to measure the capacity of storage devices, but H can obtain non-integer values.

7.4.2.2 Joint entropy

The entropy can also be calculated for a combination of two or more discrete random variables (a vector). For two variables X and Y, the joint entropy $H(X, Y)$ of the vector (X, Y) is defined as

$$H(X, Y) = -\sum_{i}^{N}\sum_{j}^{M} p(X_i, Y_j) * log_2 p(X_i, Y_j). \tag{7.3}$$

Note that the equation

$$H(X, Y) = H(X) + H(Y) \tag{7.4}$$

holds if, and only if X and Y are independent.

7.4.2.3 Conditional entropy

The entropy of Y given X is called average conditional entropy H(Y/X). It is defined as

$$H(Y/X) = -\sum_{i}^{N}\sum_{j}^{M} p(X_i) * p(Y_j/X_i) * log_2 p(Y_j/X_i). \tag{7.5}$$

It relates to the joint entropy as

$$H(X, Y) = H(X) + H(Y/X). \tag{7.6}$$

7.4.2.4 Mutual information or information content

The information content of two variables $I(X;Y)$ tells, how much information is in X about Y and *vice versa*. It is defined as

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(Y) - H(Y/X) \tag{7.7}$$

As can be proved from equations 7.2 to 7.7, the information $I(X;X)$ of a random variable about itself is the same as its entropy $H(X)$.

To achieve comparability between information values, they should be normalized, resulting in the maximum redundancy $R_{max}(X;Y)$:

$$R_{max}(X;Y) = \frac{I(X;Y)}{min(H(X), H(Y))} \tag{7.8}$$

To analyze the data in Columba, the set of 22 properties of PDB entries from all data sources was used. For each property, the Shannon entropy was calculated according to equation 7.2. For each pair of properties X and Y, the normalized mutual information or maximum redundancy $R_{max}(X;Y)$ according to equation 7.8 was calculated.

7.5 Ways to access data in the Columba database

7.5.1 Direct queries using the Structured Query Language (SQL)

The SQL language, used common by all RDBMS, is a powerful formalism to quickly retrieve information from a database. Therefore, a number of database access tools has evolved, ranging from simple command-line editors (psql) to full-featured window-based environments (Aqua Data Studio). All of them have basically the same purpose: They let the user enter and submit SQL queries, which always return data in tabular form. In this study, both types of software were used for preliminary studies on the data, manual checks on its consistency and for debugging.

7.5.2 Indirect queries through scripting languages

When the questions to be answered by a database get more complex, finding aproppriate SQL queries becomes difficult. Although SQL allows one to combine and hierarchically arrange queries, and even allows one to write functions through script-like features, this is not always efficient. Using a common scripting language, most of which provide database interfaces, the same result can be achieved in much less coding time. A script, submitting multiple simple SQL queries and subsequently combining the results, will often solve the task in the same time while providing maximum flexibility. The choice, whether pure SQL or which scripting language are used, depends on the users' expertise in either of them.

Where not specified, database queries were performed in combined Python/SQL code throughout this study, especially for the construction of sample datasets.

7.5.3 User-friendly queries through the Columba web interface

Submitting SQL queries or writing a script is only feasible, if the database user has an user account on the RDBMS server. Such access for large numbers of anonymous users is reasonable, mainly for security concerns and for the relatively high complexity of SQL. Some databases, like MSD [Velankar et al., 2005], provide very advanced interfaces, in

which users can build their own queries by combining search conditions in a graphical point-and-click interface. However, this kind of interface is quite complex both for the user and the maintainer, while still not covering all of the options that SQL and scripting languages offer. Therefore the decision was made not to implement this kind of approach.

A Python CGI script on the web server creates a SQL query from the user input, submits it to the database, and the result comes back as a HTML page. In the web schema of the database, the queries of multiple simultaneous users are deposited. There, the session ID, IP address, host name and login time for each Columba user session are stored. Also, each query submitted by a web user is stored.

In the Columba web interface, one or several data source-specific forms are presented to the users, where they can enter terms they are looking for. The input from the forms is used to find matching entries from the PDB. In this approach, the user needs to decide, which of the integrated databases and which fields to query.

Additionally, it was important to have full text search capabilities, like Google, PubMed and many other web sites have. As the name implies, full text search looks for entries from a database that contain a given keyword anywhere in their text items. PostGreSQL offers the possiblity to use the full text indexing module 'tsearch2'. The full text search in PostGreSQL only works on a single table. Therefore, an extensive table, containing all textual fields from all tables in the columba schema (the normalized representation), was created.

Individual contributions to the Columba database

Kristian Rother was responsible for the data model, the implementation of the annotation workflow, filling the database and analyzing the data. Silke Trissl designed and implemented most of the web-interface, developed procedures for performing efficient queries. Heiko Müller helped importing PDB data and supported the data model design. Stefan Günther imported CATH data. Raphael Bauer imported the BioSQL data and implemented the full text search. Philipp Hussels included the JMol plugin in the website. Thomas Steinke maintained the server infrastructure. Ina Koch and Patrick May provided data on the topological classification of protein folds. Robert Preißner provided data on protein ligands. Elke Michalsky took part in the analysis of completeness and redundancy of data. Ulf Leser supervised the development of the database and the website continuously. Cornelius Frömmel coordinated the project and provided sample questions in different development stages.

Several people, including all of the above, took part in carefully testing those parts of the project they did not implement themselves.

data source ID	database	description	priority
pdb	PDB	Protein structures	1
biosql	GO	Ontology terms that are organized hierarchically	1
biosql	Swiss-Prot	Protein sequences and annotation of sequences	1
boehringer	Boehringer	Metabolic pathway maps	2
cath	CATH	Hierarchical protein fold and domain classification	2
dssp	DSSP	software that calculates secondary structures	2
enzyme	ENYZME	Description of enzymatic functions	2
goa	GOA	Links from GO to Swiss-Prot	1
kegg	KEGG	Metabolic pathways	2
ncbi	NCBI Taxonomy	Taxonomy of most known organisms	1
pdb_clusters	PDB-Clusters	Clusters of homologous chains	2
pdbsprotec	PDBSprotEC	Links from Swiss-Prot to PDB	3
pisces	PISCES	Non-redundant lists of protein structures	2
ptgl	PTGL	Protein topology graphs	2
scop	SCOP	Hierarchical fold classification	2
systers	Systers protein family server	Protein families from hierarchical clustering.	2
biosql_mapping	-	Links biosql to pdb	2
fts	-	Provides full text search	5
goa_mapping	-	Links goa to biosql and pdb	4
ncbi_mapping	-	Links ncbi to biosql and pdb	4
web	-	Prepares database for web server	6

Table 7.1: Data sources included in Columba. In the upper section, data sources that integrate biological databases are listed. The data sources in the lower section are required for managing the data inside the database once it has been integrated to query it efficiently. The priority indicates the order in which the data sources are to be processed (compare to figure 7.4). The web links for each source database are listed in section D.

schema	# tables	description
columba	30	The main body of protein structure annotation.
biosql	31	**Swiss-Prot** sequences and **Gene Ontology** terms (see A.2.2 and A.2.2).
goa	1	**Gene Ontology Annotation** for Swiss-Prot entries (see A.2.7).
ncbi	31	**NCBI taxonomy** (see A.2.9).
ptgl	5	The **PTGL** Protein Topology Graph Library (see A.2.13).
public	4	Indices for full text search.
web	6	Session data and queries from the web interface (see 7.5.3).

Table 7.2: Database schemas in the Columba database.

8 Packing of distinct regions in protein structures is quantified by an improved Voronoi procedure

In this study, packing in three types of structures was examined: In a reference set containing high-quality protein structures of all types, the membrane domains, and in sites for small molecule-binding. All three structural groups were analyzed using an improved Voronoi procedure [Goede et al., 1997]. Special attention was given to interior cavities and their neighboring atoms. Their location relative to the protein surface was analyzed. First, the datasets used in this study will be presented. Second the atom types that were distinguished will be defined. Third, the packing measures that were calculated will be explained.

8.1 Datasets used for atom packing analysis

8.1.1 Representative protein structures

The reference set ensures comparability of the other datasets and to previous studies [Tsai et al., 1999, 2001], and it was used to draw conclusions relevant for all heavy protein atoms. A representative set of 87 high-resolution protein structures (resolution $<$ 1.75Å, R-value $<$ 0.20) each representing a distinct SCOP superfamily was suggested by [Tsai and Gerstein, 2002]. Homologous chains were removed from each dataset, and the first alternative conformation was used, if more than one occured. In table 8.1, the PDB codes in the dataset are listed. To compare packing in different regions of the interior, the atoms in this set have been subdivided into several sets depending on their distance to the protein surface. This method takes non-spherical protein shapes into account. Average packing densities have been calculated for different atom types within each group.

8.1.2 Membrane protein structures

Membrane channels and transporters have been grouped together under the term *membrane gates*, because of the regulated opening or closing of the pore underlying their function [Hildebrand et al., 2004]. On the other hand, proton pumps, receptors, and photosystems can be classified as *membrane coils*, because the helix interaction motifs are

reference dataset							
193l	1aac	1amm	1arb	1bpi	1cbn	1chd	1cka
1csh	1gai	1lcp_A	1poa	1tad_A	1xyz_A	2erl	
1cse	3ebx	1fnb	1krn	1php	1sri_A	1xso_A	2eng
1ctf	1ctj	1cus	1cyo	1edm_B	1epn	1ezm	1fkd
1gof	1hms	1igd	1isu_A	1jbc	1kap	1knb	1kpt_A
1lit	1lkk	1llp	1mla	1mol_A	1mrj	1pdo	1phc
1ptf	1ra9	1rge_A	1rie	1rop	1rro	1smd	1snc
1tca	1tfe	1thw	1utg	1vcc	1vhh	1vsd	1whi
256b_A	2act	2bbk_HL	2cba	2cpl	2ctb	2dri	2end
2ilk	2olb	2ovo	2phy	2sil	2sn3	2trx_A	2wrp
3cla	3grs	3pte	3sdh_A	4fgf	5p21	7rsa	8rxn

Table 8.1: PDB-codes of the reference dataset. The 87 structures were taken from [Tsai and Gerstein, 2002].

membrane gates									
1eul	1iwg	1j4n	1jvm	1kpl	1l7v	1msl	1okc	1pw4	1rh5
membrane coils									
1aig	1c3w	1ezv	1f88	1jb0	1kqf	1nek	2occ	1q16	1qla

Table 8.2: PDB-codes of membrane proteins sets collected by Hildebrand [Hildebrand et al., 2004].

very similar to those characteristic of soluble coiled-coils [Langosch and Heringa, 1998]. It was demonstrated that the members of these two groups differ markedly in torsion angles and intrahelical hydrogen bonds [Hildebrand et al., 2004]. This observation fostered the hypothesis, that both groups would also show characterisic packing densities.

Accordingly, a non-redundant set of structures of helical transmembrane domains was manually compiled from the PDB by Hildebrand [Hildebrand et al., 2004], using wild-type structures with the highest resolution (see table 8.2). Nonetheless, the structures of membrane coils are resolved at a higher mean resolution than membrane gates.

In these membrane proteins, only the interfaces of transmembrane helices were used for calculation. The membrane boundaries were assigned manually. Helical interfaces were defined as pairs of molecular surface patches between neighboring helices that are in direct contact with each other. Direct contact means that the atomic Van-der-Waals surfaces are closer than a given cut-off distance [Preissner et al., 1998]. It was previously shown that molecular surface patches are generally flat atom assemblies with a length/width/depth ratio of $3 : 2 : 1$.

8.1.3 Small organic molecules

Little is known about packing of binding sites and ligands. The main reason for this is that most small molecules reside at the protein surface, a location where the analysis of packing properties encounters fundamental problems. Fortunately, protein ligands are found in multiple orientations in different proteins, thereby bringing one side at a time into the protein interior.

The Columba database was used to create a dataset of structures binding small molecules. In table 8.3, a sequence of conditions that were applied to the database is listed in pseudocode. First, all available hetero groups were considered. 627 types of hetero groups that occur in the PDB more than once (90% of all groups) were retrieved. These were manually assigned to one of 7 classes, for instance *ions*, *metabolites* and *coenzymes and prosthetic groups*. After water, the most frequent hetero groups in the PDB are selenomethionine (15,794), magnesium ions (9,450) and GlucNAc (6,019).

It was jugded neither reasonable to analyze metal ions nor integral parts of polypeptide chains. Instead, 15 frequently occuring coenzymes and 16 small organic compounds were manually chosen for further analysis. As many chemically different coenzymes as possible were selected, for which sufficient numbers of copies were available. The small molecules were included in order to determine whether their size impedes an analysis.

From the PDB data, a set of structures containing these ligands was retrieved using the Columba database (see table 8.3). Within each structure file, identical chains, alternative conformations and water molecules were eliminated. From the resulting data, only hetero groups with at least six structures were used. The remaining 21 hetero groups are listed

hetero groups
- Select name and number of structures for all hetero groups in the PDB. - Manually select interesting groups. - Select all entries from the PDB that a) have X-ray crystallography as experimental method. b) have a resolution <= 2.0Å. c) are in different 50% sequence identity clusters according to [Li et al., 2001]. d) contain at least one of the hetero groups with the codes ACN, ACT, ACY, BME, EDO, EGL, EOH, FMT, GOL, MOH, SEO, FAD, FMN, GSH, GTT, HEM, NAD, NAI, NAP, NDP and PLP.

Table 8.3: Conditions applied to the Columba database to create a dataset of coenzyme/small ligand-binding protein structures. The conditions are given in a schematic language that can be easily translated to SQL code. The complete list ofthe PDB-codes in this dataset are listed in appendix C.

in table 8.4.

8.2 Definition of atomic subsets in protein molecules

To define different regions in the protein interior, the distance to the Connolly surface was used. In a second approach a higher number of concentric shells were defined by their distance to the next atom on the Connolly surface. The Connolly surface is defined by the surface of a spherical 1.4Å-radius probe rolling over the protein atoms' Van-der-Waals spheres. Protein atoms were assigned to atomic subsets, according to the shortest distance $dsurf$ between the center of an atom and the Connolly surface [Connolly, 1983]. Ligands and other non-peptide atoms were included. All water molecules at the protein surface were removed, because the extent to which they are resolved differs greatly between individual structures. Seven atomic subsets were defined (see figure 8.1):

1. SURFACE - This set contains all atoms that are directly reached by the probe, and, therefore, they are in contact to the Connolly surface ($dsurf = 0.0$Å).

2. SHALLOW - Contains all atoms no more than 2.8Å apart from the Connolly surface but not come into contact with it (0.0Å$< dsurf <= 2.8$Å).

3. DEEP - Contains all atoms more than 2.8Å apart from the Connolly surface ($dsurf > 2.8$Å).

4. BURIED - This is the union of the SHALLOW and DEEP subsets ($dsurf > 0.0$Å).

5. BOTTOM - Defined similarly to the BT set in the analysis of [Tsai and Gerstein, 2002], a separate Connolly surface is calculated for the BURIED subsets' atoms. This set consists of all atoms from the BURIED subset that are not in contact with that surface.

6. CAVNB - This set contains atoms in direct contact with a cavity, as explained below.

code	ligand	N(structures)	N(ligands)	N(in PDB)
small organic compounds				
ACN	acetone	7	14	33
ACT	acetate ion	146	355	995
ACY	acetic acid	79	220	565
BME	β-mercaptoethanol	48	106	485
EDO	ethylene glycol	100	580	1163
EGL	ethylene glycol	43	266	558
EOH	ethanol	18	70	171
FMT	formic acid	43	201	366
GOL	glycerol	344	1276	3594
MOH	methanol	6	31	208
SEO	β-mercaptoethanol	10	22	272
coenzymes and prosthetic groups				
FAD	flavin-adenine dinucleotide	61	107	707
FMN	flavin mononucleotide	45	74	328
GSH	glutathione	10	24	63
GTT	glutathione	12	23	75
HEM	protoporphyrin IX	126	242	2717
NAD	nicotinamide adenine dinucleotide	65	146	869
NAI	1,4-dihydronicotinamide adenine dinucleotide	6	10	71
NAP	nicotinamide adenine dinucleotide phosphate	43	88	362
NDP	1,4-dihydronicotinamide adenine dinucleotide phosphate	24	42	264
PLP	pyridoxal-5-phosphate	45	85	603

Table 8.4: Hetero groups used for packing analysis of binding sites. The leftmost column gives the residue codes for each hetero group, as used by the PDB. The number of ligands is the number of molecules from this hetero group in the final dataset, while the rightmost column gives the number of ligands for the entire PDB.

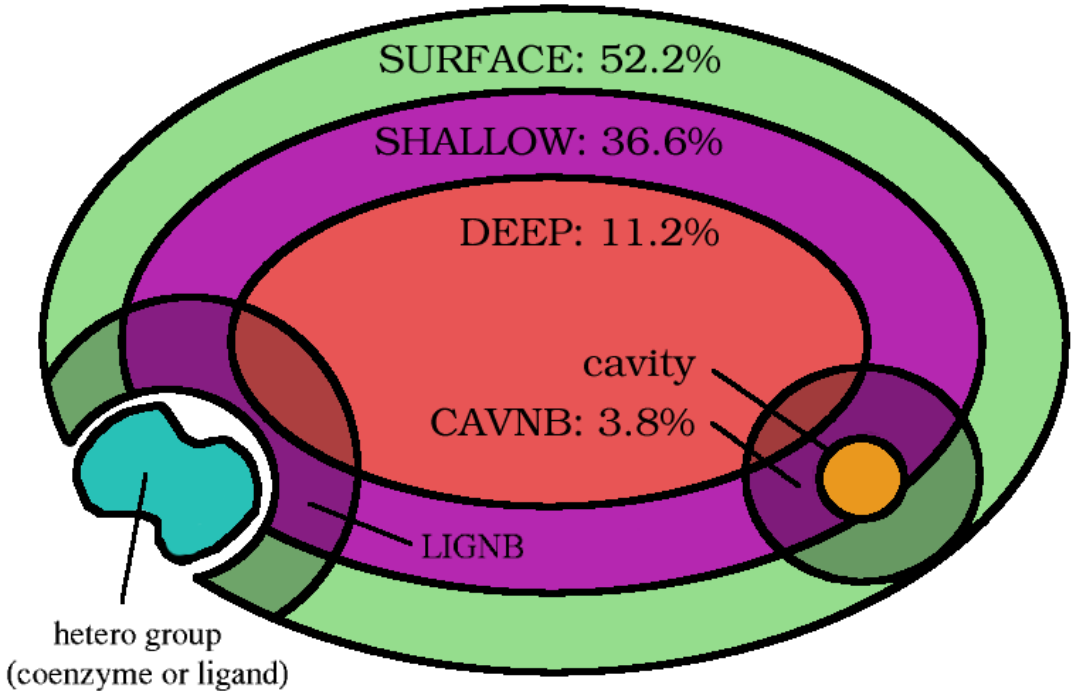

Figure 8.1: Schematic definition of atomic subsets. The subsets SURFACE (green), SHALLOW (purple), DEEP (red), LIGNB and CAVNB (both shaded) are distinguished. An example cavity position is marked by the orange sphere. The SURFACE subset was defined as all atoms at the Connolly surface, the SHALLOW subsets' atoms were between 0.0 and 2.8Å apart from and the DEEP subset was below that. Cavities were detected using a 1.4Å radius probe.

7. LIGNB - This set contains atoms in direct contact with one of the hetero groups from table 8.4.

8.2.1 Definition of atom types

To apply the Voronoi procedure to volume calulation, each atom needs to be assigned a Van-der-Waals radius. In total, 173 different atom names occur in polypeptide chains, including oxidized and reduced cysteine. Terminal oxygen atoms were ignored, because they are ambiguously denotated in many PDB-files. Clustering, by chemical and numerical criteria results in 18 atom types with similar properties [Tsai et al., 2001]. These atom types were named by the scheme ExHyZ, in which E is the element, x the number of covalently linked atoms, y the number of hydrogens and Z is a letter distinguishing between big and small subtypes of a particular chemical type (see table 8.5). Each atom was assigned a Van-der-Waals radius from a set of radii, empirically determined from structures of small organic molecules, known as the ProtOr set [Tsai et al., 1999]. Radii for non-protein atoms were determined separately [Stouten et al., 1993]. The term *atom*

ProtOr type	radius [Å]	description
C3H0	1.61	aromatic carbon (branched)
C3H1	1.76	aromatic carbon (unbranched)
C4H1	1.88	branched alkyl carbon or functional group (e.g. amide)
C4H2	1.88	methylene group or any primary carbon
C4H3	1.88	methyl group
N3H0	1.64	aromatic nitrogen or tertiary amide
N3H1	1.64	amide nitrogen (e.g. peptide bond)
N3H2	1.64	amino group
O1H0	1.42	carbonylic or carboxylic oxygen
O2H1	1.46	hydroxyl group
S2H0	1.77	thioehter or cystine bridge sulfur
S2H1	1.77	thiol group
N2H0	1.64	aromatic nitrogen
N4H0	1.64	quarternary ammonium (charged)
O2H0	1.42	ester-, ether- or acetalic oxygen

Table 8.5: ProtOr radii for all atom types used in this study. Twelve types (in the upper part of the table) have been adopted from [Tsai et al., 1999]. Three types (in the lower part of the table) occur only in some of the hetero groups, and radii for them have been adopted from one of the other types.

groups is also used for atoms with the ProtOr typing scheme, because properties of a heavy atom and its linked hydrogen atoms are combined to one structural entity.

For each of the 21 hetero groups, a standardized nomenclature of atoms exists in the PDB, defining topologically unambiguous *chemical atom positions* within a molecule. Hetero groups, not following this nomenclature or with missing atoms were excluded from the dataset (less than 3%). Based on chemical expertise, the appropriate ProtOr atom types out of the 18 different types were manually assigned to each chemical atom position in each hetero group. All chemical atom positions were treated independent throughout this study. In some cases, the initially published ProOr atom types [Tsai et al., 1999] did not suffice, and a few new atom types were introduced (see table 8.5). The radii for them were adopted from the next most similar atom type, a practice that has been employed and proven unproblematic by [Voss and Gerstein, 2005]. For some elements, like phosphorus, the radii provided by Stouten [Stouten et al., 1993] were used. This set of radii was also used for all calculations on the membrane protein dataset.

8.3 Calculation of local atomic packing densities

The local packing density of an atom, quantifies, how close an atom is located to its neighbors. To calculate it, atomic volumes are determined to weigh each neighbor atoms' contribution in a reasonable way. For this, a modified Voronoi procedure was used, which allocates the total space within a protein structure among all atoms by constructing hyperboloid interfaces between them [Goede et al., 1997]. The interfaces are defined by all points that are equally distant from both Van-der-Waals spheres of the two atoms (see figure 8.2).

At the protein surface, it is not practical to distribute all available space among

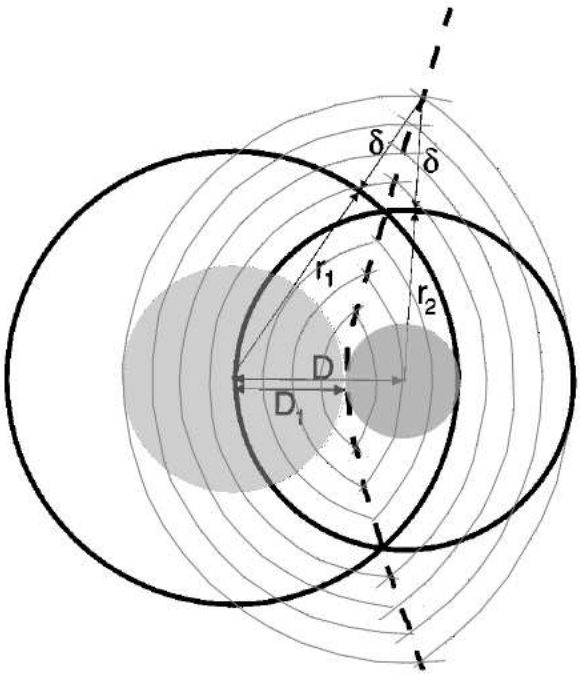

Figure 8.2: Construction diagram for the improved Voronoi procedure [Goede et al., 1997]. The dividing hyperboloid surface between two atoms is constructed as follows: Two spheres with the Van-der-Waals radii r1 and r2 are drawn around both atoms' center points. Both radii are extended by $\delta \in \Re$. All points where the lines $r1 + \delta$ and $r2 + \delta$ extending from the atom centers meet are defined as the dividing surface of these atoms.

PDB-code							
1dxr	1qov	1aij	6prc	4prc	1dv6	1prc	1e14
4rcr	1pss	1k6l	1l9j	1jh0	1dv3	1aig	1pst
3prc	1ds8	1pcr	1jgx	5prc	1mps	7prc	2prc

Table 8.6: PDB-codes of 24 structures from the photosynthetic reaction center. They were used to elucidate whether the packing density depends on the crystallographic resolution.

neighboring atoms [Gerstein and Chothia, 1996]. Therefore, the maximum distance from the atom center to any point, belonging to that atom group, is defined as the Van-der-Waals radius plus 1.4Å representing a solvent molecule. In this manner, the volume of an atom can be divided in two (see figure 5.1). The Van-der-Waals volume of an atom, V_{VdW}, is the space inside the atom's Van-der-Waals sphere, cut by the separating surface between covalently linked atoms. The solvent excluded volume of an atom V_{SE} is also cut by separating surfaces. The local packing density PD can be calculated from both of these, which is defined as the fraction of both values [Finney, 1970]:

$$PD = \frac{V_{VdW}}{V_{VdW} + V_{SE}} \tag{8.1}$$

Atom group volumes were numerically calculated, using a cubic lattice with a grid distance of 0.2Å. For each atomic subset and atom type k in the reference set, the average packing density $< PD^k >$ was calculated as being the mean value of the local packing densities of all contributing atoms having the standard deviation σ^k. The same was calculated for the transmembrane domains and the hetero group dataset. For the latter, each chemical atom position in each hetero group was calculated independently (e.g. data for C3H1 aromatic carbons in different positions in the flavin molecule were not mixed).

To qualify whether the resolution could influence an analysis - especially for the transmembrane domains - the packing density values, of 24 different structures in the photosynthetic reaction center, were compared. The PDB-codes are listed in table 8.6. As a result, it was observed that the average packing density ($< PD >= 0.81$, $\sigma = 0.01$) does not depend on the crystallographic resolution (between 2.1-3.5Å).

To compare the packing of a distinct protein with average values, the packing z-score-rms Z_{rms} introduced by Pontius [Pontius et al., 1996] is defined:

$$Z_{rms} = \sqrt{\frac{1}{n}\sum_{i}^{n}(\frac{PD_i^k - < PD^k >}{\sigma^k})^2} \tag{8.2}$$

8.4 Identification and localization of atom-sized cavities

In this study, cavities were defined as locations, at which a 1.4Å-radius probe could be placed inside a protein structure. The probe inside a cavity must not intersect any atoms' Van-der-Waals sphere and the cavity must not extend to the protein surface [Richmond, 1984]. To estimate the positions of cavities in the protein interior in relation to the distance to the surface, all protein atoms were divided into 20 layers of atoms. For each protein atom, the distance to the next surface atom *sdist* was calculated. All atoms in the BURIED subset were subdivided into 20 disjoint groups characterized by the *sdist*

ranges such that each group contained the same number of atoms. These groups were termed as shells. By this approach, the atom numbers of each shell are equal while their volumes may differ slightly due to different packing of individual atoms.

The centers of the cavities were calculated as an average from the atom coordinates of the cavities' neighbor atoms, ignoring the mass of different atom types. These points will be referred to as *cavity positions*. Each cavity position was assigned to one of the 20 shells by its *sdist* value. For this part of our analysis, cavities occupied by one or more water molecules were treated as if they were empty.

The relative amount of cavity positions and cavity neighbors for each shell was calculated for the reference dataset. As this is not reasonable for the other datasets, only the absolute numbers were analyzed there. Finally, the frequency of polar and nonpolar cavity neighbor and other atoms in each shell was calculated. All nitrogen, oxygen and all peptide/amide carbon atoms were defined as polar. These carbon atoms have a partial charge similar to most oxygen and nitrogen atoms due to the strong electronegativity of their neighboring atoms [Brooks et al., 1983].

Part III

Results and Discussion

9 In the Columba database, two thirds of the entries in the Protein Data Bank are well-annotated

The Columba database is a relational, integrated database of information on protein structures. It is designed to support the creation of sets of protein structures based on their annotation. Columba integrates sixteen databases containing different aspects of protein sequences and structures. For their integration, sixteen corresponding data source modules have been implemented, each consisting of a data model and a data workflow part. Each data source is considered as a dimension, in which PDB entries are annotated. This approach has been termed multidimensional data integration (published in [Rother et al., 2004]), which was inspired by data warehouse design [Chaudhuri S, 1997]. Five additional data sources have been designed for administrative tasks. The resulting data model of Columba is called a star schema, with the PDB in the center and the other data sources linked to it. It contains 6 database schemas, 101 tables, 6 views, 139 indices, 35 sequences and 17 functions (see section 7.2, terms explained in appendix E).

Columba was filled with data using the annotation workflow, described in section 7.3, consisting of about 8,600 lines of code, not including third-party parsers, modules and scripts. The annotation workflow runs approximately 48 hours on a 1.8 GHz computer for a complete filling of the Columba database. The database dumpfile resulting from this is 1,770 MB in size. Individual contributions from the developers to the Columba database are listed in section 7.5.3.

9.1 Quantity and quality of the data in Columba

All 30,960 entries from the Protein Data Bank (as of June 2005) were imported to Columba, and data from the other data sources was added upon this import (published in [Trissl et al., 2005]). Figure 9.1 lists the numbers of entries for each table in the *columba* schema, the main part of the database. These differ considerably, ranging from 72 non-redundant PDB subsets from PISCES and 155 manually curated KEGG pathways up to 4,089,006 entries in the *tax_graph_path* table that links all species from the NCBI taxonomy with their ancestors. The data from the PDB is found in the four tables *pdb_entry, compound, chain* and *hetero* (see section 7.2.2).

9.1.1 Entries from the PDB and from secondary databases

The PDB archive forms a global stockpile of protein structures. There are no releases, as is common to many other databases, and the protein composition is highly non-uniform. According to SCOP superfamilies, the four most common proteins were already composing 9% of the entire database. Precisely, these are: 820 lysozyme-like PDB entries, 791

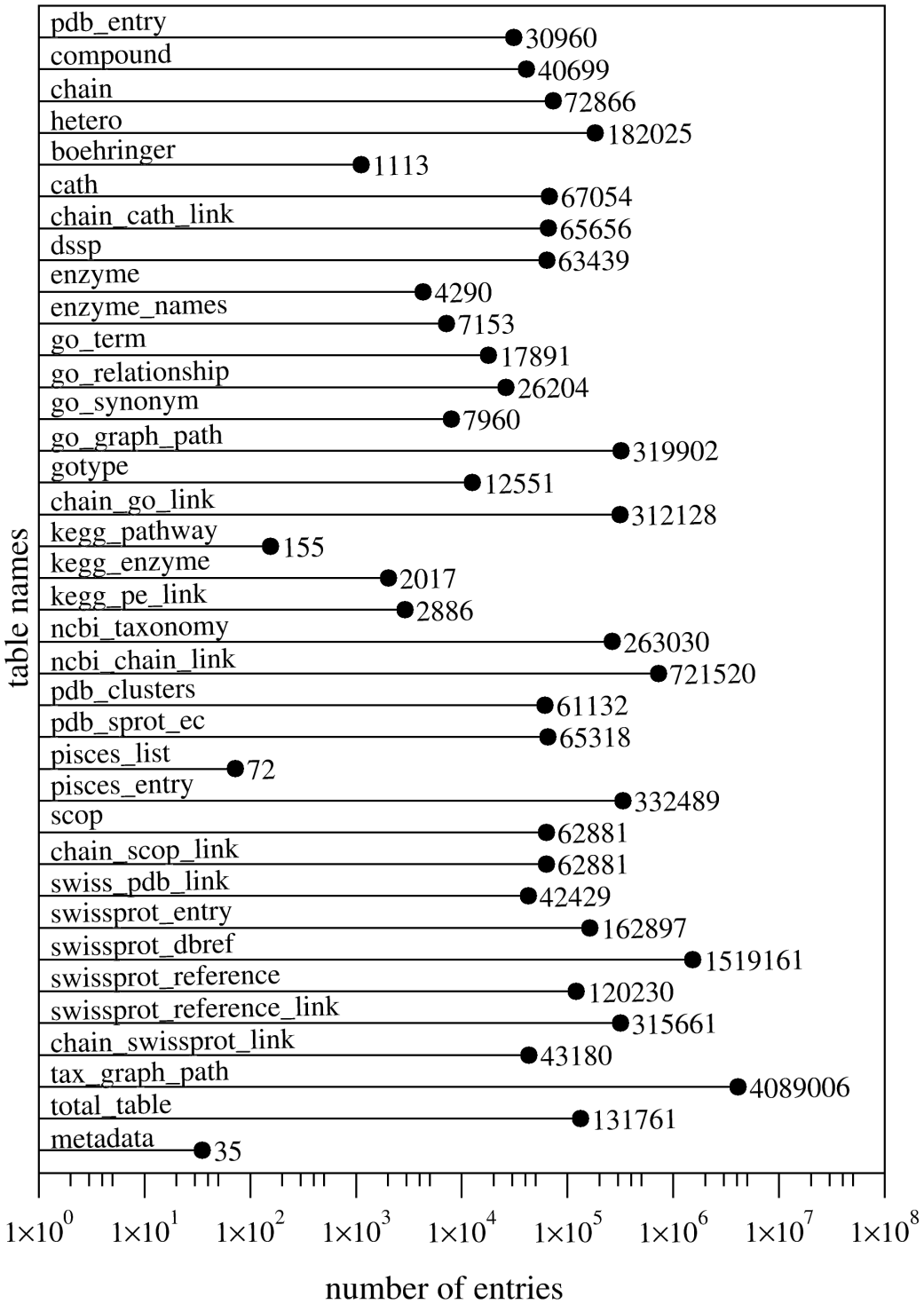

Figure 9.1: Number of entries in all tables of the Columba schema.

immunoglobulins, 712 trypsine-like proteases and 420 globins. In total, 1,395 entries were found that did not contain a polypeptide chain (pure nucleic acids, cyclic D-peptides and other small molecules).

It is worth noting that Columba does not store the coordinates of structures themselves. Columba is designed to enable users to find a specific set of structures based on annotation, but not to analyze the structures themselves. This is intentional, since extensive software that can efficiently parse, visualize, or compare protein structures from PDB files already exists. Also, the inclusion of structural data would increase the size of both the data model and the database by, at least, one order of magnitude. Representation of the structural data from the PDB in a relational database has been implemented by the OpenMMS [Greer et al., 2002] and MSD [Velankar et al., 2005] projects.

In the PDB, both structural and functional aspects of proteins, protein domains and important sites are only marginally represented. The non-uniform representations of these data have resulted in a poor data quality. The fields for which standardized data exists (PDB-id, method, resolution, r-value, date and textual description) are represented in Columba. Annotation from the other integrated databases was connected to each PDB entry by the PDB identifiers.

9.1.2 Folding classification and secondary structures

SCOP [Murzin et al., 1995] and CATH [Orengo et al., 1997], two hierarchical fold classifications, have been included in Columba, both of which are maintained by experts, supported by automated classification systems. SCOP contains classification on four levels for all PDB structures except non-peptidic molecules. CATH annotates only proteins that naturally occur as folded domains. Both of them provide definitions of domains within each polypeptide chain. In CATH, the lower hierarchy levels have numbers instead of names. While this unambiguously identifies them, it is in some respect impractical for the human user.

For each structure containing a polypeptide chain, secondary structures were calculated with the DSSP program [Kabsch and Sander, 1983]. Because DSSP is the quasi-standard for secondary structure definition, the results of its calculations are unquestionable in this context. A sequence of secondary structure codes (H for helix, E for extended, G for 3/10-helix etc.) was built for each protein chain. These sequences should have exactly the same length as the amino acid sequence in Columba. This was the case for 63,439 sequences, which were stored in Columba. In 3,609 additional sequences, non-standard residues and badly labeled alternative conformations caused inconsistencies between the two sequences, and they were discarded.

9.1.3 Sequence data

The Swiss-Prot database [Boeckmann et al., 2003] was fully included, allowing the annotation from GO/GOA [Camon et al., 2004] and the NCBI taxonomy database [Wheeler et al., 2000] to be linked to Columba. Links from Swiss-Prot entries to PDB entries and chains were gained from the PDBSprotEC database [Martin, 2004]. While a Swiss-Prot entry describes the same protein as a corresponding PDB entry, the amino acid sequences from both databases are almost never the same: of 43,180 references from Swiss-Prot entries to PDB chains, only 4,007 have identical sequences. This is due to missing N- and C-termini, missing loops, artificial point mutations or uncrystallized domains, which affect

many PDB structures. However, all PDB chains almost perfectly match sub-sequences of the Swiss-Prot entries. The GO terms and the GOA annotation provide unambiguous assignments of functions to sequence entries and thereby also to the PDB.

Some features of Swiss-Prot sequences like, *'extracellular location'*, are not given for all datbase entries and therefore difficult to query. Other important properties of proteins, such as *'can be phosphorylated'*, *'regulates or inhibits another protein'* are found neither in the sequence entries nor in the GO annotation. It would be desirable that the annotation would be more detailed here.

The organisms assigned to proteins via the NCBI taxonomy are uniform and, in this way, are more reliable than those listed in the PDB. Because the taxonomy links are established via Swiss-Prot entries, there is no risk that the expression system is accidentally annotated instead of the real source organism. For these reasons, the NCBI taxonomy should be used instead of the information in the PDB whenever possible.

9.1.4 Descriptions of enzymatic and metabolic function

Enzyme descriptions from the ENZYME catalog [Bairoch, 2000] provide manually curated functional annotation. Of the 1,161 different E.C. numbers used in PDB entries, all but 22 were contained in the current ENZYME release. These 22 included 7 obsolete E.C. numbers and 15 typographical errors. Also, it is known that several enzymes have more than one E.C. number; it is not certain that every protein structure can be associated to all biologically relevant reactions by only matching E.C. numbers. Columba also contains the coordinates on the Boehringer Biochemical Pathways poster [Michal, 1993] for 1,113 enzymes, allowing them to be found rapidly.

The metabolic pathways from KEGG [Kanehisa et al., 2004] contain many manually curated links to enzymes. The data has been integrated perfectly, but the user needs to know some peculiarities within KEGG: i) not all proteins are enzymes and not all enzymes occur in pathways; ii) the pathways in KEGG include not only main metabolic routes and regulatory paths, but also sub-pathways, side-pathways, and anaplerotic reactions; iii) the actual pathways in a specific organism often deviate considerably from a generic pathway definition in KEGG; thus the genetic/proteomic configuration of that organism needs to be checked. The level of detail in Columba could be improved by integrating more data on metabolic compounds, organisms and genes from the KEGG project.

The links to ENZYME, KEGG and the Boehringer maps could be improved using the links provided by PDBSprotEC [Martin, 2004]. The EBI is about to refurbish this database soon, and therefore it would make no sense adapting to PDBSprotEC before then.

9.1.5 Non-redundant subsets of the PDB

The PISCES representative lists contain lists of PDB entries, from which redundancy has been removed by various conditions [Wang and Dunbrack Jr, 2003]. However, there is a complication: representative entries will be selected by PISCES only if they are not filtered out by another filtering condition beforehand. PISCES knows no second-best structure that it could use instead. This drawback is avoided by the clustering, done by Li [Li et al., 2001], on the PDB server. Here, all protein chains are ranked within clusters, with 50%, 70% and 90% identity, allowing the creation of a non-redundant dataset, by taking the highest-ranking structure from each cluster after all other filter conditions have

been applied.

The SYSTERS database [Krause et al., 2000] also assigns protein chains to clusters for many of the proteins in Columba. While the underlying hierarchical clustering principle is certainly superior to a simple Smith-Waterman sequence identity threshold, it is difficult to maintain SYSTERS within the Columba database continuously, as the SYSTERS database is not available in a downloadable format. For these reasons, representative sets created with PISCES, pdb_clusters and SYSTERS will rarely be identical.

Extensive efforts are presently being undertaken by the PDB maintainers to improve the data quality of PDB by standardizing the annotation [Berman et al., 2003]. It can be expected that many follow-up projects of the PDB, like Columba, will profit from this development.

9.2 Completeness of secondary annotation on the PDB

Initially, it was studied to what extent the PDB entries are covered by the other databases. Considering this is important as database cross-references are often likely to be outdated, wrong or incomplete. To evaluate the integrated database, the interconnections between the data sources also were be analyzed.

The cross-references for twelve of the sixteen data sources were compared to the full set of PDB entries and also compared to each other. In addition to the data already in Columba, a few additional databases were included in the analysis to improve comparability of the data: BRENDA, UniProt and InterPro, as described in section 7.1.1.

9.2.1 Time-dependency of secondary annotation

It has been observed that coverage of many data sources is time-dependent (see figure 9.2, 9.3 and 9.4). The amount of available secondary database annotation has dropped considerably for structures released in the recent few years. This time-dependent annotation gap can be observed for many of the second-party databases (published in [Rother et al., 2005]).

As can be seen in figure 9.2, the coverage of all fold/family databases but HSSP declined from 1995 to 2000. HSSP - which is calculated automatically for each new PDB entry- remains constant throughout the years. SCOP stays at a high coverage for a long time, followed by CE, DALI and CATH. The HOMSTRAD project has not kept pace with the recent growth in data. Even though there were manual updates in the beginning of 2003, HOMSTRAD covers less than 40% of the PDB entries.

For sequence databases, the situation is similar (see figure 9.3). Structures deposited after the mid 1990's are affected with respect to Swiss-Prot and UniProt. For InterPro and PDBSprotEC, coverage drops only after 2001, where the drop is less severe for PDB-SprotEC. It must be pointed out that the curves for both pairs Swiss-Prot/UniProt and PDBSprotEC/InterPro are to a certain degree similar, suggesting that these databases share their data, at least partially.

The curves for the ENZYME and KEGG databases (figure 9.4) show that the amount of enzymes in the PDB has remained constant for about ten years. In the 1980's, enzyme classification was enforced less strictly by the PDB maintainers, and the amount of links was smaller then. BRENDA has failed to annotate some datasets of enzymes published since 1998; PDBSprotEC has missed only a few since 2001. Comparing the latter two with the ENZYME links, which were created using the E.C. numbers from the PDB, it

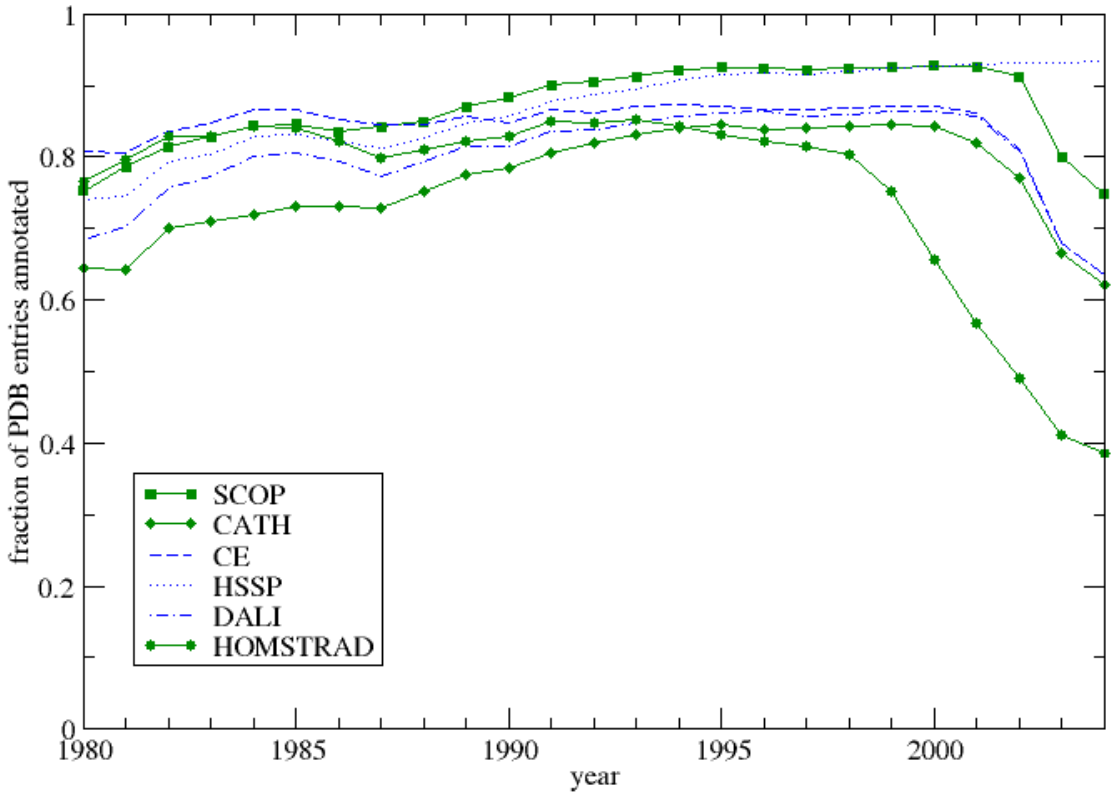

Figure 9.2: Fraction of PDB entries that are annotated by fold classification databases for each year. During the 70's, the PDB was very small and large variation is observed. After 1997, the data sources where references require manual interaction were not able to keep pace with the rapid development of the PDB.

becomes clear that the slight decrease in the BRENDA and PDBSprotEC curves is really due to a lack of database links and not due to a drop in the relative amount of enzymes in the PDB.

9.2.2 Coverage of the PDB by secondary databases

The fraction of the PDB annotated by the second-party databases varies considerably. As mentioned above, one cannot expect that databases reach 100% coverage over each other. For instance, there will never be links from ENZYME to non-enzyme structures in the PDB; CATH does not consider disordered structures or peptides - while SCOP does, and Swiss-Prot excludes immunoglobulins. The maximal coverage of data sources thus varies greatly, depending on the nature of a database and the policy of its maintainers. However, computing a maximal PDB coverage for each secondary database is very difficult. In the following section, an effort will be made, wherever possible, to contrast the absolute coverage with the informal thoughts on the expected maximal coverage.

HSSP comes the closest to approaching full PDB coverage (93.5%), followed by SCOP (73.0%). CATH, DALI and CE all cover approximately 63%. These five databases cover mostly the same structures, and 55.3% of the PDB are covered by all of them. Of all PDB entries, 73.7% are linked by the Swiss-Prot sequence database, which is almost the

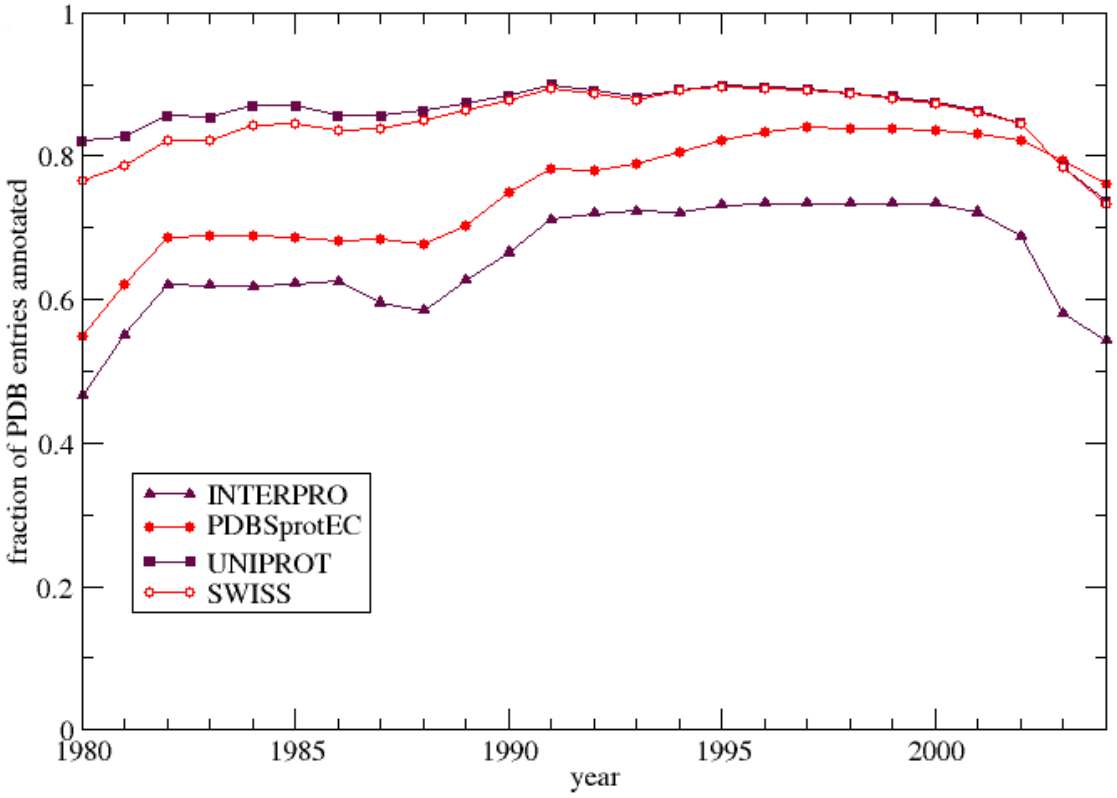

Figure 9.3: Fraction of PDB entries that are annotated by sequence databases for each year.

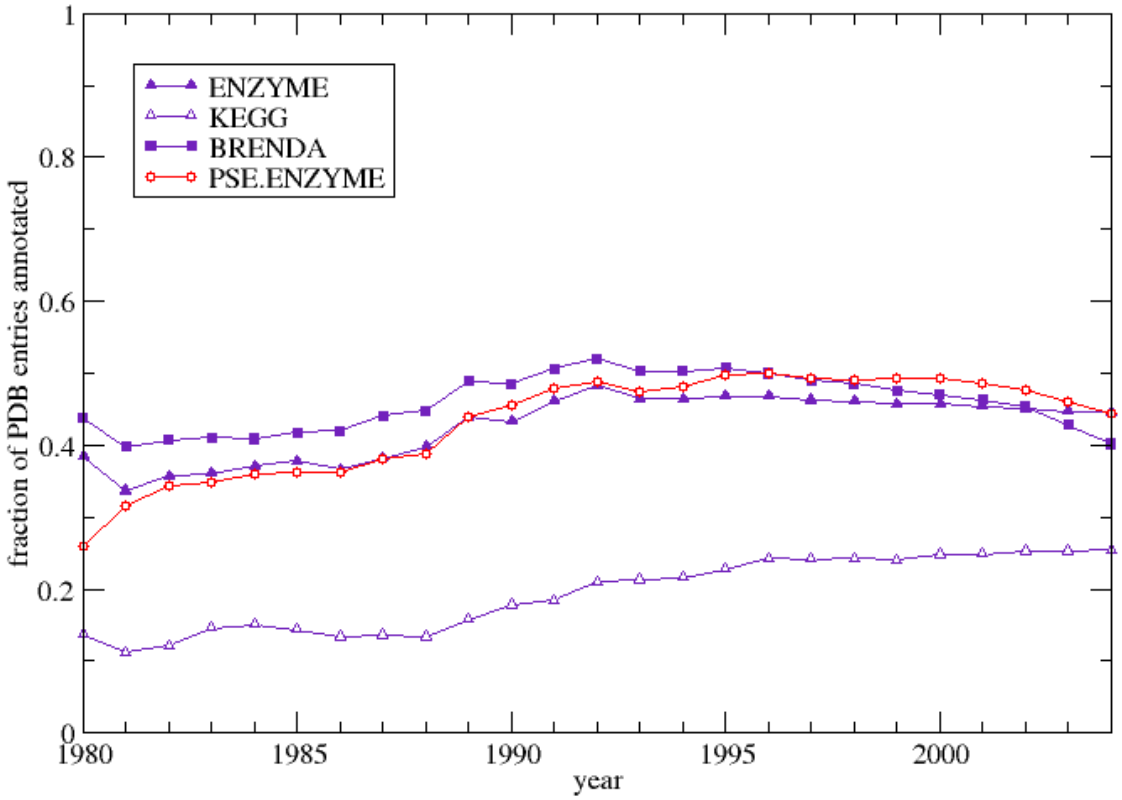

Figure 9.4: Fraction of PDB entries that are annotated by enzymatic function databases for each year.

same amount of links as UniProt. However, by using PDBSprotEC (76.2%), this amount still can be increased. InterPro contains much fewer references (54.3%), which can be explained by the fact that InterPro only links to sequences containing certain domains. Two thirds of the structures (65.7%) were annotated by PDBSprotEC and either SCOP or CATH, suggesting that these data sources overlap to a very high degree.

Additionally, the coverage of two databases annotating Swiss-Prot entries, which in turn are linked to the PDB via PDBSprotEC, was analyzed: functional annotation from the Gene Ontology (GOA [Camon et al., 2004]) and taxonomic classification from the NCBI [Wheeler et al., 2000]. Both of these annotate (indirectly) almost as many structures as their corresponding sequence database entries. This demonstrates how important it is to have high-quality Swiss-Prot references.

Almost half of the PDB structures (13,296) are designated as enzymes. Of these, 92.2% have an E.C. number, indicating a specific enzymatic function described in the ENZYME database. BRENDA annotates 83.1%, and PDBSprotEC annotates 91.6% of the enzymes in the PDB. Only 52.4% of these enzymes could be linked to the detailed pathway descriptions from KEGG. The enzymes missed by ENZYME, BRENDA, and PDBSprotEC have been judged as enzymes according to the first lines of the PDB header, not the E.C. numbers. These three sources annotate 70.3% of all enzymes.

There were 11,054 structures (40.2%) annotated by all of the following databases; SCOP, CATH, HSSP, PDBSprotEC, UniProt, InterPro, GOA, NCBI taxonomy, and, for enzymes, ENZYME and BRENDA. In contrast to that, 2,196 structures are annotated by only HSSP or ENZYME/KEGG, most of them being recent entries. 1,462 structures were found to have no secondary annotation at all, including 1,357 nucleic acid structures, 41 small molecules, 52 proteins (partially with unknown function) and 12 low-resolution structures.

9.2.3 Comparison of databases

To display the interdependencies between the data sources, the overlap of any two sets of linked PDB entries was calculated as the number of entries at the intersection of both sets divided by the number of entries in their union (see equation 7.1 in section 7.4.1. Based on these overlaps, we computed a tree using the UPGMA method [Sneath and Sokal, 1973]. The tree illustrates the degree of overlap in the content of the different databases (see figure 9.5). The entire overlap matrix is shown in table B.1. It becomes clear that the fold/family classifications and the sequence databases group well together. Only those data sources having comparatively high (HSSP) or low coverage (HOMSTRAD and InterPro) are located in different regions. The enzyme-related data sources are located on a different branch, since they have much fewer links and do not overlap much with the other sets. The two additional sources, GOA and NCBI, are almost identical to the PDBSprotEC links that were used to link them.

Finally, it was analyzed whether representative sets of structures available on the web show a better coverage by secondary annotation than the average PDB entry. The representative structures offered by the PDB and from the PISCES server were compared to randomly created subsets of the PDB having the same size. Surprisingly, it was found that random structures are annotated by an average of 9.5 data sources of the 16 considered, whereas the representative structures are annotated by only 9.0 (PDB) and 9.3 (PISCES) data sources. This may be due to the fact that both methods prefer more

recent structures in the representative sets, which tend to be less well annotated.

9.2.4 How well are PDB entries annotated in secondary databases?

Protein structures deposited in the PDB before 1997 are generally well-annotated: the fraction of entries covered is constantly around 90% for SCOP and HSSP, 80% for several other fold/family classifications, around 80% for sources containing Swiss-Prot references and 40% for enzyme databases. The score for SCOP is higher than the score for CATH, since SCOP contains classifications for peptides and non-standard proteins not considered by CATH. These figures are probably the highest possible, given that PDB also contains structures that are not addressed by these databases, such as non-proteins and synthetic

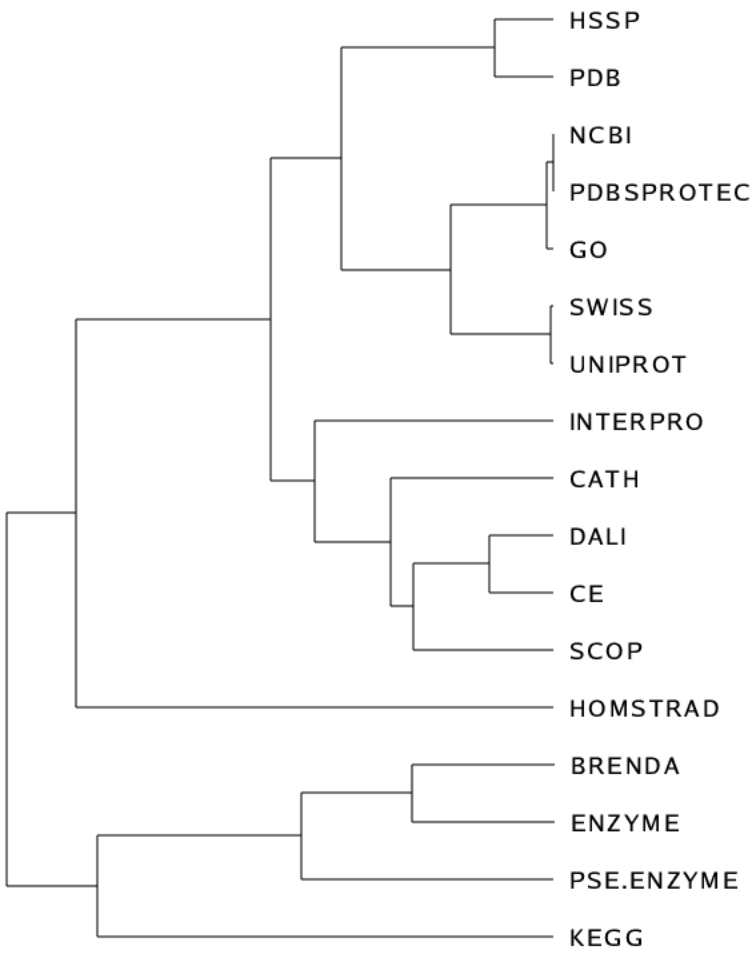

Figure 9.5: Tree of data sources constructed using the overlap score described in the text. The best-overlapping data sources were assigned a common node first. Besides each node, the overlap score for all data sources below that node is given on the left side, on the right side the percentage of the PDB covered by that nodes' sources. The overlap matrix is shown in table B.1 in the appendix.

peptides. DALI and CE are similar to SCOP.

Roughly two-thirds of the PDB entries are linked to both a sequence database and one or more fold/family classifications. This coincides with earlier results on the coverage of fold classification databases [Hadley and Jones, 1999], suggesting that the availability of annotations has, in this respect, not changed considerably in the last years. References to ENZYME or KEGG are more frequent after 1990, because the PDB started to enforce submission of E.C. numbers from that time on. After 2000, references to the folding classifications, except HSSP, became less abundant (see figure 9.2). The same applies for sequence-related databases and BRENDA from 1997 on. Note that this affects a time range, within which about 75% of the PDB entries fall. Classification by HSSP, ENZYME and KEGG is available even for very recent structures, since it is calculated automatically. For 1,462 structures, no annotation at all exists yet, but they are mostly nucleic acids and other non-proteins.

9.3 Redundancy within the data quantified by Shannon entropy and maximum redundancy

It has been discussed above, that a large amount of annotation exists for many protein structures, and that the data is interconnected well. But how many different proteins do they really describe? It was shown in 9.1.1 that the four most frequent protein families already compose 9% of the PDB. Another important issue is whether the source databases are different or whether they only describe the same thing in different ways.

Using methods from information theory, both questions can be quantified: The Shannon entropy is employed to quantify how redundant a particular textual field within the database is, whereas the maximum redundancy tells how similar two of these fields are.

9.3.1 Shannon entropy of properties of PDB chains

For the former, 22 columns of data from all over the Columba database were retrieved, containing textual data and discrete numbers. These 22 columns will be termed properties. A list of all PDB chains, having a particular property, was created with the corresponding values. For instance, the property *ncbi_taxon* produced a list of organisms for the chains in the PDB. For each of these lists, the Shannon entropy has been calculated according to equation 7.2.

As is observed in figure 9.6, most of the 22 properties are a little below the maximum possible entropy. Some of them are well below the maximum value, indicating that these properties are highly biased. The *pdb_method* (84% X-ray, 15% NMR, 1% others) contains almost no uncertainty. The highest entropy was found in *cluster90* (the identifier of families with 90% homology). In all cases, the composition of the PDB, not of the secondary data, is responsible for the level of redundancy that is visible as the difference between observed entropy and maximum entropy. These calculations have been performed per PDB-chain. Calculating entropies for these properties on a per-structure basis leads to similar results (data not shown).

9.3.2 Maximum redundancy of properties of PDB chains

For many pairs of the 22 properties introduced above, it is clear that they correlate in some way: e.g. protein chains in a cluster with 90% sequence identity will still cluster at 70% identity. In addition, the levels of the SCOP and CATH hierarchies also depend on

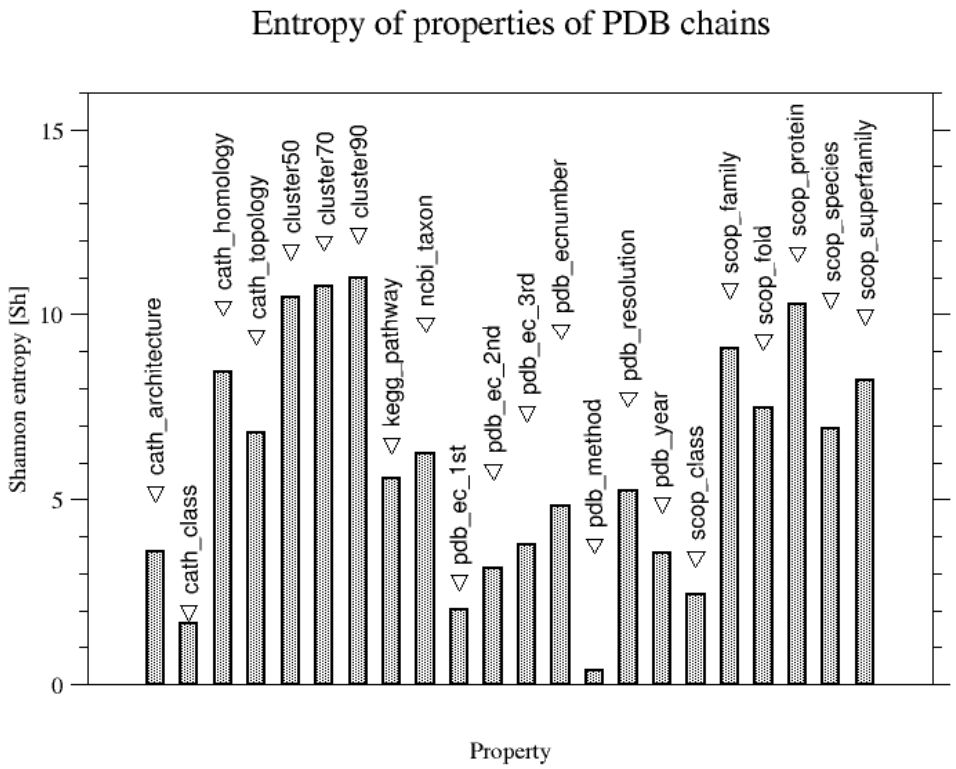

Figure 9.6: Entropy of 22 properties of PDB chains. The bars indicate the entropy that the fields have. The diamonds indicate the maximum entropy that property would have if it was perfectly randomized (this relates directly to the number of classes. The unit in which information is measured is Shannon (Sh) or bit.

each other. But to what extent do the corresponding levels of SCOP and CATH correlate? How about non-obvious combinations like E.C. number and experimental method?

First, all pairs of properties were tested to find out whether they were statistically independent. Using the χ^2 test (α=0.05), any combination produced a significant correlation owing to the redundancy of the database.

To elucidate how much information one property carries about another, the mutual information 7.7 was calculated for each pair. These values were normalized by the smaller entropy of the two properties in a pair, resulting in the maximum redundancy $R_{max}(X;Y)$ (see section 7.4.2, equation 7.8). The sum of all $R_{max}(X;Y)$ values for one property was calculated. Figure 9.7 displays how much information one property carries about all others.

As an initial observation, the information about many properties having a low entropy is almost fully contained in a few high-entropy properties: Knowing a protein's 90% homology family allows one to predict properties like the E.C. number or the protein fold almost perfectly. On the other hand, there is not much information about properties derived from the PDB like *resolution*, *year of deposition* and *method for structure determination* available in the other fields.

The maximum redundancy is always within the interval $(0;1)$. Thus the maximum for the sums in figure 9.7 is 22. Owing to the composition of the PDB, the properties defining protein families (*scop_family*, *cath_topology*, and *cluster##*) in a high level of detail carry much information about the others. In fact, some properties like *cluster90* and *scop_protein* get very close to the maximum. This means not only that the annotation in the databases is redundant to a very high degree, but also that the properties do not differ very much among each other.

It follows that many of the correlations shown above would disappear if a non-redundant dataset were used - depending on the criteria for non-redundancy. Using such a dataset, it was found that the protein fold does not correlate with the enzyme class (first digit of the EC number), but on the type of ligand it binds [Orengo et al., 1997]. Other correlations would remain, for instance the correlation between a sequence identity cluster and a SCOP fold. Another important conclusion is that non-redundant datasets, generated using sequence identity, SCOP family or CATH family as a criterion, will not differ much. Only for recent structures, the fold classification is often missing. Therefore, the sequence identity calculated from the PDB chains as done by Li [Li et al., 2001] is certainly a superior method to removing redundancy.

9.4 Discussion of the implementation of the Columba integrated database

9.4.1 Implications of the star-shaped data model

The database schema in Columba is called a star schema, in correspondence with the visual appearance of the tables. These are holding information from protein structures in the center of a set of tables containing the data from other data sources. This approach never merges logically similar types of information into a single table. In contrast, a tighter semantic integration was for instance followed in the TAMBIS [Baker et al., 1998] project. Columba is built upon the strong belief that merging data from different databases into the same tables would be misleading for the biologist. To avoid methodically different content becoming mixed, each included data source was assigned its own set of tables. On the other hand, keeping data separated inevitably leads to a certain degree

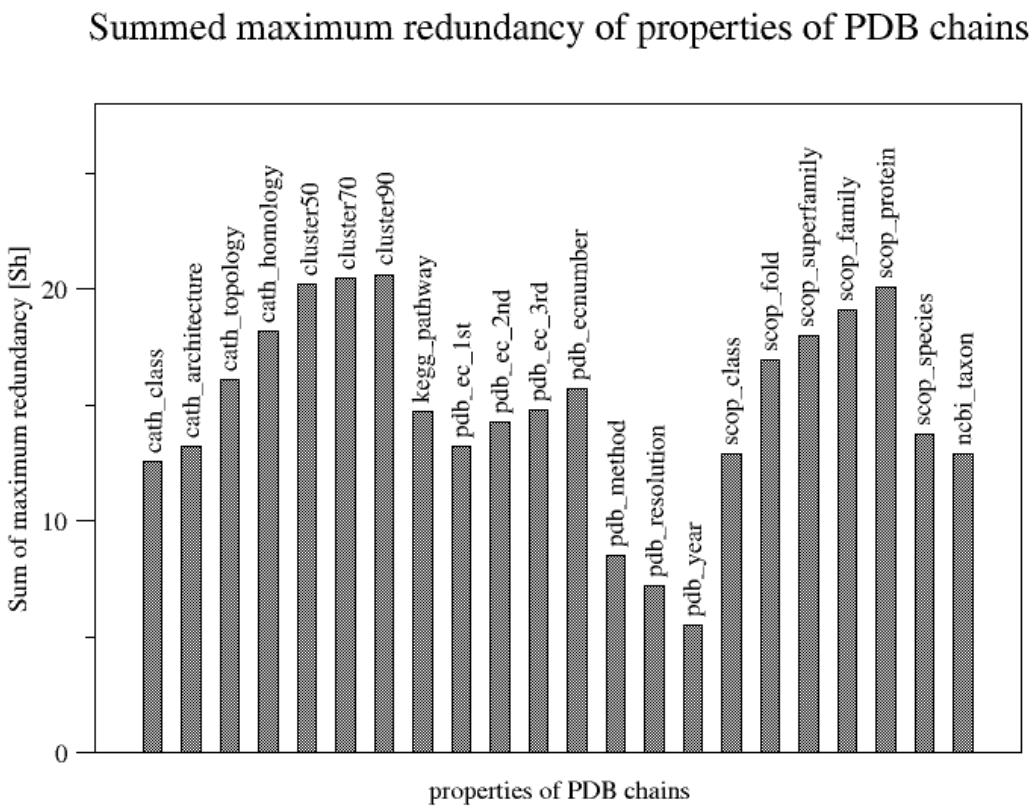

Figure 9.7: Sum of maximum redundancy for 22 properties of PDB chains. The maximum redundancy is the mutual information for a pair of properties, normalized by the smaller of the entropies of the two properties. As the information is normalized to values in the interval (0;1), the sum for all 22 pairs has a maximum of 22.

of semantic redundancy: i.e., different schema elements provide the same type of information. For instance, functional annotation of proteins is encoded both in Swiss-Prot keywords and Gene Ontology terms; TIM barrels are annotated in CATH, SCOP, and the PDB annotation itself, but the definition and naming of domains can be quite different [Hadley and Jones, 1999].

Therefore, the redundancy does not originate from data duplication alone, but rather from evidence obtained independently by different people or by different experiments. The advantages of this approach prevail for mainly two reasons: First, users recognize the origin of the data they query and obtain as result. From personal experience, biologists often have their favorite set of databases, about which they know the pitfalls and peculiarities. By keeping data separated, personal preferences or differences in trust in particular databases can be expressed and the results can be judged based on prior experience. Second, subtle differences in the semantics of fields of different databases are conserved. For instance, both Swiss-Prot keywords and GO annotations express functional annotation. However, the process of creating this annotation is quite different, and it is often worthwhile to discriminate between the two. Furthermore, separating data and software for the different data sources greatly simplifies system maintenance. Changes to data sources, including the deletion or addition of data sources, only affect a well defined part of the schema and of the web interface.

Various approaches of data integration, using multiple schemas and subsequent mapping, exist. Regarding schema design [Paton et al., 2000] suggests conceptual models for different types of Life Science data. However, these models are not designed for data integration, and no model is proposed for protein structures. Considering annotation sources as dimensions describing some primary objects is also an important aspect of the EnsMart project [Kasprzyk et al., 2004]. EnsMart uses a reversed star schema to connect genes with different types of information, such as genomic position, transcription factors, or expression data. The data is queried through a generic web interface, which also allows source-specific queries and their combinations. Conceptually, EnsMart and Columba are very similar, but they work on totally different types of data. Moreover, Columba is directly designed for handling annotations of protein structures, which has advantages in terms of result visualization and search options.

Another advantage of integrating data into a relational database is that specialized software can detect relationships between data items automatically. Recently, the *Aladin* approach (Almost automatic data integration) was developed [Leser and Naumann, 2005]), which could generate a completely cross-referenced data warehouse with a minimal effort in building it.

Representing graphs and networks in a relational database is a special challenge (see section 4.2). In Columba, this was achieved by the introduction of extensive index structures that link a node within a graph to all its supernodes (for the NCBI taxonomy and GO terms). Alternative approaches have been compared by Trissl, resulting in an extended pre- and postorder ranking scheme that is applicable to directed acyclic graphs [Trissl and Leser, 2005].

9.4.2 Implications of the annotation workflow architecture

In order to keep data from different sources separated, modularization was judged more important than a data model optimized to accelerate queries. Assigning each table to

one data source greatly enhances maintainability. The performance of queries is often not that important when doing analyses (it does not matter much whether a query takes 15 or 50 seconds) compared with the ability to write many queries in a short time. It does matter, however, in a case, in which the relational database offers several possibilites to support queries over several tables (indices and materialized views), which are used by the Columba web interface (see 9.5.1).

PDB entries are central to the data model but not to the workflow. Many data sources turned out not to depend on the information that comes with PDB entries. For them, the filling process required only the call to a single procedure. Nontheless, the dependencies of data sources on each other need to be designed carefully. As mentioned above, they can be derived from the data model, and the dependence diagram in figure 7.4 is basically a simplified representation of the data model itself.

The runtime of the annotation workflow is important. The major constraint of the projects advance was development time. This involved, of course, extensive testing which became a limiting factor several times because of the comparatively long runtime of particular data source modules (*biosql, biosql_mapping, goa_mapping*!). Owing to the modularized architecture, all data sources could be debugged independently. This was more difficult the longer a data source module took to be filled. Therefore, two approaches were proposed to shorten runtime: i) The data was written to ASCII tables that could be loaded rapidly via the PostGreSQL *COPY* command instead of a large number of *INSERT* statements (used for the CATH data source). ii) Intermediate results were stored outside the database (used for files calculated by DSSP, and for calculated PDB-Swiss-Prot alignments).

Changes in the model of the source databases were not encountered. The *database churn*, reported by Stein [Stein, 2003] that thwarted the Integrated Genome Database (IGD) [Ritter et al., 1994], was not a problem in the Columba project. There were two cases, in which the software needed to be adjusted because of the data, and both were owing to errors in the source data files. Thus, it is assumed that the parsers used for Columba work very reliably.

9.5 The Columba database is made available via a web interface, dump files, and third-party software

Columba solves all three kinds of challenges in data integration (see section 4.1): It delegates the semantic problem to the end user, but gives him powerful technical means to handle the data conveniently. To make the Columba database usable for other scientists (and thereby address the sociological issue), three different approaches are being used. These are i) using the web interface, ii) transferring database dump files, and iii) integrating Columba with other software.

9.5.1 Querying Columba through the web interface

Columba can be searched through a web interface available at *http://www.columba-db.de* (published in [Trissl et al., 2005] and [Rother et al., 2006]). The interface allows two types of queries: Full text search, and data source and attribute specific searches. In both cases, the search results in a list of PDB entries and their corresponding chains.

Full-text search interface. For the sake of convenience and as a quick-start, Columba can be searched by using a standard keyword search of all textual fields in Columba (see

Figure 9.8: Main page of the Columba website. The input field in the lower part of the screen allows for use of the full-text search feature of Columba immediately.

Figure 9.9: Input form of the Columba website, allowing constraints to PDB entries to be entered. The buttons to the left provide similar forms on other annotation sources. The yellow bar indicates, how many PDB entries remain when all constraints are applied cumulatively.

Figure 9.10: Result list of the Columba website. Each line corresponds to one PDB entry, including a rough description of what compounds and chains exist and what the protein is. The links on each PDB entry lead to the Columba explorer.

Figure 9.11: Columba explorer page of the website. This page contains all available annotation for one PDB entry in a structured manner.

figure 9.8), including the annotation given by the PDB, enzymatic, metabolic, taxonomic, and the protein-fold classification information. Keywords can be combined using logical AND, OR, and NOT operators. The keyword search performs a simultaneous search of the content of all integrated data sources, and is, thus, a quick and easy-to-use option for finding interesting protein structures. However, it does not allow for source- or attribute specific queries.

Data source-specific queries. To support queries such as *finding all protein structures annotated in CATH as Rossmann fold*, a web interface based on the paradigm of query refinement was created . This process is best understood as having an initial dataset, which is subsequently reduced by applying different filters. The initial dataset contains the entire set of PDB entries. For each of the data sources integrated into Columba, the user may specify source-specific filter conditions using a proper web form (see figure 9.9). The source-specific forms can be found by using the labeled buttons on the left side of the web page. After entering conditions in a form, those PDB entries that do not fulfill the stated conditions are removed from the current set of results.

Several forms can be applied consecutively, thus restricting the original set of all PDB entries by conditions placed on multiple data sources. Conditions on different sources are always logically connected by an AND. The available search operators depend on the specific field and data source, ranging from numerical comparisons to substring matching and traversal of ontological structures. To guide the user, Columba constantly shows the current number of qualifying PDB entries following each query step in the header of the page. This demonstrates the consequences of adding, deleting, or changing conditions and helps to prevent the over-specification of search conditions, which can lead to empty sets. Note that the full-text search can be used as an additional restriction condition on the result set, which has turned out to be a quite powerful feature of the search interface.

Results page and structure explorer. Once the user has specified all desired conditions, Columba computes the matching set of protein structures. This list of results (see figure 9.10) gives basic information, such as PDB ID, experimental method, compound name, and chains for each entry. The PDB entry ID links to the Columba Explorer view for that particular entry. The Explorer (see figure 9.11) shows all information stored in Columba for that PDB entry. This includes the experimental method and resolution for each entry and compound name, metabolic information, and the source organism for each compound. Detailed information is given for each chain, including protein-fold classification from SCOP and CATH, data from the according Swiss-Prot entry, Gene Ontology annotation, and NCBI taxon name. Links to the original data items in the respective databases are also provided.

To further enhance the search capabilities of the web interface, it is possible to upload a file containing a set of PDB identifiers. These identifiers are used as an additional filter condition on the result set. Thus, a user can view all data in Columba for the entries in his list and create subsets of protein structures from the list by entering conditions on second-party annotations. Thereby, the Columba web interface greatly reduces the time required to collect additional information for entries in any list of PDB entries.

Additional features of the web interface It possible to link to the Columba explorer pages to show a single PDB entry directly. The URL is
*http://141.20.27.240/columba/columba.cgi?action=single_result&pdbid=****xxxx***,

where *xxxx* is the PDB-code of the according structure. Both the query results and reports for single PDB entries can be retrieved in XML format. The latter provide the complete annotation from Columba in a machine-readable way. In order to access them from scripts and third-party applications, the URL
*http://141.20.27.240/columba/columba.cgi?action=single_result_xml&pdbid=**xxxx***
can be used. To display the 3D-structures of proteins on the website, the Java-based molecular viewer JMOL is being used to display 3D-structures of the proteins in the web browser.

9.5.1.1 Ideas for futher improvement of the web interface

Unification of terminology. Inconsistent use of terminology currently limits the searchability of the PDB. The PDB uniformity project [Bhat et al., 2001] and the Macromolecular Structure Database MSD [Velankar et al., 2005] both aim at correcting PDB entries, unifying terminology by using a precisely defined XML dialect [Westbrook et al., 2005], and adding or updating links to scientific references. The MSD database also addresses the linkage of PDB chains to Swiss-Prot entries. Thus, it can be expected that these efforts will improve the applicability of Columba in the near future, for instance, if the PDB entries themselves come with consistent and structured taxonomic information.

Advanced query options. The public access could be improved further by i) combining sub-queries by operations other than *AND*, ii) providing a fully mature XML-DTD (document-type definition), iii) offering some kind of web service, like those in the Distributed Annotation System (DAS) [Dowell et al., 2001]. Ultimately, the scientific community would profit the most from accessible flat file dumps of the complete database the most, but this is severely hindered by licensing restrictions.

Estimation of the reliability of queries. Finally, a data warehouse integrating many data sources, like Columba, provides the opportunity to assess its own query results. It is hypothesized that a Columba entry containing a search term in many data sources is more likely to be the desired result than an entry containing it in only one. By counting how often a query term occurs in the full annotation of an entry from the result set, a 'reliability score' for each result entry could be calculated.

9.5.2 Database dump files

A relational database is able to export the data structures including the content as a single SQL file. These can be imported by an RDBMS onto a different machine. It is convenient to distribute a database in this way, but a number of problems need to be considered: First, the data size (1770 MB for a complete dump) is likely to discourage potential users. Second, Columba dump files are specific for the PostGreSQL RDBMS and will be incompatible to other database servers like MySQL, Oracle, or DB2. Third, providing a complete dump of the Columba for download will conflict with the licensing conditions of some constituent databases. In contrast to the software used for Columba, the data is free for academic usage only, imposing constraints on its usage.

For these reasons, dump files of Columba are not being and will not be made available to the public. However, they have proven reliable for backup/restore operations and for transferring data between different RDBMS servers within the Columba development team. Alternatively, solving both the technical and legal issues, single tables from the database could be made available as zipped ASCII files.

9.5.3 Embedding Columba in other software projects

At the moment, four bioinformatic applications interact with the Columba database. The protein sequence alignment editor StrAP [Gille et al., 2003] provides links to the Columba explorer pages to get a quick overview on structures, for which structural alignments are being made. Similarly, the molecular viewers PyMOL (www.pymol.org, via the rtools extension www.rubor.de/bioinf) and BRAGI [Reichelt et al., 2005] access the Columba website and show results for one given PDB structure. Additionally, BRAGI uses the SCOP domain boundaries stored in Columba to highlight protein domains. Recently, the Raptor3D software has been published, mapping annotation from multiple databases, including Columba, into protein structures [Dieterich et al., 2005].

9.6 To create a protein structure dataset, seven questions need to be answered

The observations made in sections 9.2 and 9.3 allow a general description of the PDB and its annotation, on which queries are to be built upon: The majority, roughly two thirds, of the PDB entries are annotated by at least a sequence database and a fold classification database. A considerable proportion of the database is made up by proteins for which very many structures exist, or which are closely related. There is also a high number of proteins for which only one or a few structures exist.

In addition to this, there is a number of structures (about 23% of the PDB), for which annotation is missing. The main reason for the lack of annotation is that secondary databases need to catch up with the rapid growth of the PDB (see figure 9.2 to 9.4). Another 10% cannot be excepted to be annotated, because they are not proteins, but peptides, antibiotics, nucleic acids and the like.

To learn how both efficient and reasonable queries can be formulated, and to demonstrate the capabilities of Columba, several sets of protein structures have been produced from the database. Some of these are based on questions from current research projects, others have been designed to highlight a particular aspect of the Columba database.

9.6.1 Sample structure datasets

Eight datasets were created, directly accessing the database via an SQL editor. Some of them can also be adressed using the Columba website (9.6.7, 9.6.2, 9.6.4, 9.6.5 and 9.6.9). These queries take significantly less than a minute's time to be executed, except 9.6.4 which takes a little longer.

9.6.2 "Find extracellular proteins"

The keywords *'extracellular'*, *'secreted'* and *'attached to the outer membrane'* were searched in the comments of Swiss-Prot entries and in GO terms. The former resulted in references to 3,301 protein structures, the latter in references to 2,629 PDB entries. In both sets, 4,497 unique PDB identifiers occur, showing that the Swiss-Prot and GO annotation complement each other. Manual checks on 50 of these entries revealed no false positives. There is certainly a significant amount of false negatives, caused by missing links between PDB and Swiss-Prot, and by incomplete annotation of Swiss-Prot entries. In this type of question, it is most important to select reliable keywords for the search. A good strategy is to inspect a few known entries before starting the search.

method	no. structures	correct	false pos.	false neg.
a) Superligands (reference)	508	508	0	0
b) Andreas Hoppe	506	505	1	3
c) Columba ligands	514	507	7	1
d) Columba GO-terms	59	10	49	498
e) Columba PDB-text	237	183	54	325
f) Columba all	601	508	93	0

Table 9.1: Flavin-bound proteins identified by different methods

9.6.3 "Find proteins having the same fold but less than 25% identity"

This task was almost directly translatable from spoken language into SQL. The corresponding query in the PDB, SCOP and PISCES tables of Columba resulted in 20 protein folds, with 21 to 96 protein families (25% seq.id.) each. The folds with most families were the TIM-barrels (96), knottins (92) and immunoglobulins (71). As SCOP covers the PDB to a relatively high extent (see figure 9.2) and PISCES is updated very frequently, this query will miss no or only very few proteins.

9.6.4 "Identify seven-helical proteins"

The regular expression *'([HGI]+[BTS]+)6[HGI]+'* describes seven helices (all types) interrupted by random coils/loops, but not by sheets according to DSSP codes. A single query in the DSSP and PDB tables of Columba finds 7,127 chains in 4,124 protein structures that contain this motif. As expected, this dataset contains mostly α-helical proteins, like the whole globin family, some proteins containing β-sheets - either in a separate domain or as appendices to a helical domain - and a few membrane helix bundles.

Technically, this kind of task works perfectly. The main problem is that one has to know that more than a few proteins have very short β-strands that interrupt a sequence of helices. Using the regular expression syntax, which is explained on the Columba website, it is possible to take these cases into account.

9.6.5 "Identify flavin-bound proteins"

The goal of this task was to find protein structures that contain an isoalloxazine three-ring system, as it is found in the flavin-based coenzymes. Six approaches have been compared: a) The three-ring system was modeled as a 2D-structure, using the Marvin editor (www.chemaxon.com/marvin). With this structure, a 2D-similarity screening was performed in the Superligands database [Michalsky et al., 2005]. The resulting dataset was manually checked and used as a reference. b) A manually curated dataset from Andreas Hoppe was used. c) In the ligands stored in Columba, the regular expression **flav[(in)(o)]**, which finds both *flavin* and *flavo*, was used. The term *flav* produced huge numbers of false positives. d) The same expression was used to search the GO terms for matches. e) The same expression was used to find matches the textual annotation for PDB entries. f) The methods c-e were joined.

All retrieved structures more recent than Feb 24th, 2004 were discarded to ensure comparability with dataset b). The results are shown in table 9.1. Obviously, only the methods a)-c) produced reasonable results. Both the annotation in the PDB and in the GO terms are neither sensitive nor specific. Although methods d) and e) find the last

structure missed by c), it is not worthwhile. The seven false positives in method c) are caused by a single ligand, lumichrome which contains the three ring system but is no flavin. The false negative is proflavin, an educt of the flavins. In short, the first three methods have produced reasonable datasets.

9.6.6 "Compare culled datasets of PDB structures"

Reducing redundancy and constraining structure quality are among the filtering criteria applied most frequently to subsets of the PDB. Figure 9.12 shows how the size of subsets of the PDB using different criteria has developed over time. The main observation is that all of the non-redundant datasets show exponential growth similar to the PDB itself, only the absolute number is smaller. This implies that datasets created in the future, using the same criteria as today, will confront scientists with a higher number of checks, if they decide to review their datasets manually.

On the other hand, applying very strict criteria, like using a low sequence identity threshold to group proteins, demanding many proteins in each group, and demanding a high resolution, restricts the results to only few groups of proteins. The two lowermost curves in figure 9.12 indicate only small numbers and are growing very slowly. The resulting data belong to well-known protein families, which have been characterized in high detail. The number of different folds in SCOP and of E.C. numbers in ENZYME grow slowly, but their number is already high. This observation fits well to the data from 9.2, as both data sources are curated manually.

9.6.7 "Find biochemical pathways containing TIM-barrel proteins"

This task could be completed by combining the SCOP fold classification with the pathways from KEGG. 835 TIM-barrel proteins were found in 58 of 151 metabolic pathways. An overwhelming majority of them participate in carbohydrate metabolism pathways. This is not surprising, since these pathways are covered most extensively by structural data. Many of the enzymes found participate in more than one pathway, showing the functional diversity expressed by TIM-barrels in the metabolism.

This approach will not take all functions performed by TIM-barrel proteins into account. First, a TIM-barrel protein needs to have an E.C. number (this is the case for 1043 of 1144 PDB structures). Second, it needs to be an E.C. number listed by KEGG (this is true for 835 of the 1043 above). Both the SCOP-PDB and the PDB-KEGG references are not critical. However, one has to know that most KEGG pathways also contain numerous side reactions (see 9.6.11).

9.6.8 "Find proteins with structures from as many organisms as possible"

For structural analyses within protein families, it is desirable to have a large variety of species within the dataset. Using the data from NCBI and SCOP, it was tested whether Columba can fulfill this demand. It was found that there are 23 SCOP superfamilies for which structures from more than 20 different organisms were available. The families with the largest variety are NAD(P)-binding Rossmann-fold domains (75 species), P-loop containing nucleoside triphosphate hydrolases (58 species) (Trans)glycosidases (54 species). The other 17 superfamilies contain all the 'usual suspects' in the protein structure world. The number of structures often reaches hundreds per family - with a clear preference for structures from E.coli, S.cerevisiae and H.sapiens. The same query was tried with an

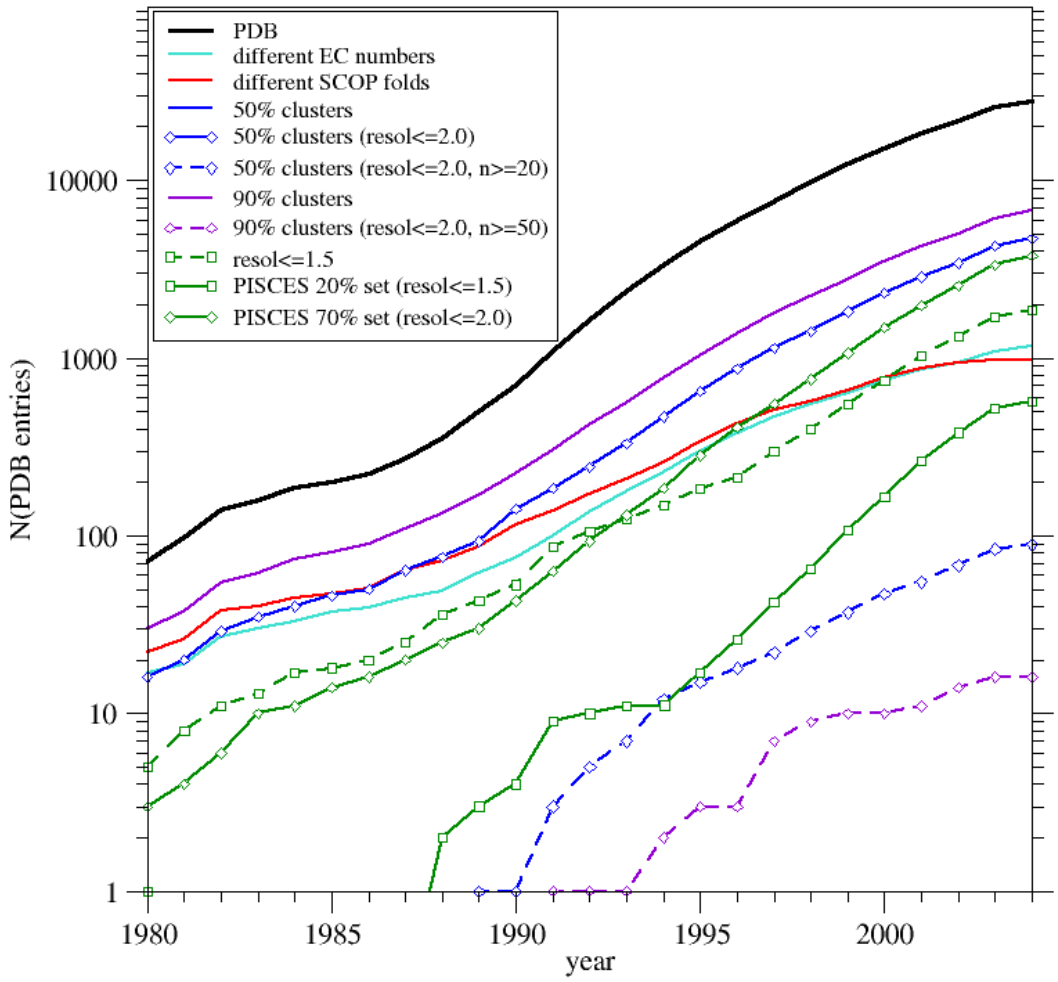

Figure 9.12: Growth of non-redundant datasets using different criteria. Each line represents the size of a set of PDB entries generated by a particular method over time. The sets labeled *xx% clusters* contain one entry per sequence identity cluster, according to [Li et al., 2001]; the *PISCES xx%* use a non-redundant list from [Wang and Dunbrack Jr, 2003]. *resol* indicates that a constraint was laid upon the crystallographic resolution, and that only X-ray structures were allowed. The n>=20/50 means that each sequence homology cluster was to contain more than 20 (or 50) structures after applying the resolution constraint.

additional 2.0Å resolution constraint, resulted in about 20% less species per superfamily. However, these numbers will decrease significantly when a higher similarity threshold is applied. It must also be taken into consideration that the intersection of SCOP fold and NCBI taxonomy annotation is only available for roughly 64% of the PDB (the two well-annotated thirds).

9.6.9 "Find proteins that relate to diphtheria"

To find protein structures related to a particular disease, structures containing 'diphtheria' were searched for in various databases (see table 9.2). Most of the queried databases agree on a core set of about 30 protein structures. These are mostly diphtheria toxins and toxin repressors. The SCOP database contains a fold *'Common fold of diphtheria toxin/transcription factors/cytochrome f'*, to which 108 protein structures belong. Most of them do not play any biological role in diphtheria but they happen to have the same fold, and are, therefore, returned by the full text search of both SCOP and Columba. Similar to the task in 9.1, this shows both the power and the dangers of full text searches. According to the organisms in the NCBI taxonomy, Columba reports back 18 structures from *Corynebacterium diphtheriae*, the microorganism causing diphtheria. These were all found by the query above.

9.6.10 Questions that Columba fails to answer

There are several types of tasks for which Columba is of limited use or an inaproppriate tool. These are discussed briefly:

"For which antigen are antibodies in the PDB specific?" Using all of the annotation in Columba, it is possible to find out the specificity of all antibodies manually. However, the information is dispersed over all of the data sources. Therefore, it is difficult to identify the names of antigens correctly, and that a particular protein mentioned is the antigen. A dictionary of protein names (which the Gene Ontology is not) or another sophisticated text mining approach would be helpful to address this task.

Database	N(PDB entries)
PDB	25
MSD	48
IMB Jena Image Library	25
SRS	31
3DInSight	30
BioMolQuest	38
SCOP	108
CATH	19
grep -r 'DIPHTHERIA' /pdb	29
Columba - PDB header	31
Columba - full text search	139

Table 9.2: Number of PDB entries related to the term *diphtheria* obtained by using various databases.

"Which proteins are modified by phosphorylation?" Columba is able to retrieve phosphate groups in protein structures. But this is neither a necessary (the structure can be un-phosphorylated) nor sufficient (the phosphate group can belong to something else) criterion for reversible phosphorylation of a protein. At the moment, the only known source for information about regulatory modifications, like phosphorylation, is the literature.

"Which proteins have glucose as a substrate?" The glucose molecules in Columba are not necessarily substrates. The KEGG database contains information about metabolic compounds and their interactions with proteins, but the data has not yet integrated into Columba.

"Are there proteins that have multiple functions?" It is known that the protein aconitase plays two important and completely different roles in the cell. Its role in the TCA cycle is well-described in Columba, but no mention of its iron-regulatory function was found. There are more proteins of this type, but it is very difficult to capture them even manually. It is imaginable that an ontology like the GO may present such facts as clearly distinct protein functions in the future.

9.6.11 General design questions for structural datasets

Obviously, selection criteria for designing representative sets of structures depend strongly on the question addressed. For this reason, it is hardly possible to solve this problem *a priori*. Instead, there are seven questions that arise for any data set that is to be created. For each of these questions, recommendations will be given below. Inspired from the sample datasets above, more specific guidelines for two types of frequently occuring queries can be formulated.

Should a subset of chains of the PDB be defined? Most secondary databases (SCOP, Swiss-Prot, PISCES, etc.) relate to chains rather than entire PDB entries. Therefore, most subsets of the PDB should be done on a per-chain, instead of a per-structure, basis. The main practical reason for this is that a structure contains multiple identical chains. Defining a subset of chains facilitates removing the redundant entries.

How is the redundancy within the PDB treated? For many datasets, reedundant entries are not desired. To get rid of the redundancy, the PDB needs to be clustered and one representative structure per cluster selected. As mentioned above, sequence similarity is best suited for the clustering, as was done by the PISCES database [Wang and Dunbrack Jr, 2003] and the PDB [Li et al., 2001]. Selecting a representative entry from a cluster should take both the quality of a structure (e.g. resolution) and the availability of secondary annotation into account to avoid the 'annotation gap' observed for recent structures. Such an approach is yet to be made.

Which resolution should be used? The main parameter that influences the size of the resulting dataset is the crystallographic resolution. For proteins, a resolution below 3.0Å can be regarded as usable, but studies concentrating on atomic details should set the threshold at least at 2.0Å. If more than one similar structure is available, the one having the best resolution should be used. NMR structures do not have a resolution, and it is difficult to determine their quality.

Is full coverage of Swiss-Prot required? About two thirds of the PDB are linked to Swiss-Prot entries. The corresponding sequence database entries not only provide annotation by themselves, but they are also used in Columba to establish links to other sources of data, such as GO terms and the NCBI taxonomy. The decision, whether to require Swiss-Prot links in all entries of the resulting dataset or not, has a high impact on the usability of the data. It makes the difference between wealth of information - or poverty, if there is one single structure in the dataset which is not linked to Swiss-Prot. For these reasons, this consideration should be made beforehand.

Should NMR structures be included? About 25% of the structures in the PDB have been obtained by NMR spectroscopy, a method very different from X-ray crystallography. Thus, they have geometrical properties (average ϕ,ψ,χ angles, bond lengths and others) deviating from the values that X-ray structures have. Also, most of them are ensembles of about 10-20 structures. In the PDB, there are also consensus structures for a particular ensemble, which have a separate PDB code for newer, but not for older structures. Comparing NMR structures with X-ray structures may provide new insights, but they will behave differently when an analysis relies on atomic details.

Are the metabolic pathways interpreted correctly? The KEGG database contains not only the main metabolic routes, but also many side reactions and organism-specific sub-pathways. On the other hand, they are not fully included in the KEGG database. Also, Columba does yet not contain the full level of detail that KEGG offers (see A.2.8). Thus, the KEGG annotation stored in Columba gives clues what kind of process an enzyme is involved in, but not about a particular metabolic route.

Will the conditions restrict the dataset too much? The coverage of data sources is far from complete. Thus, any dataset relying on a single property will filter out a portion of the PDB, especially recent structures (see section 9.2). When multiple conditions are combined, the resulting dataset quickly becomes small. Before starting, it should be considered, what the minimum number of required structures/residues/atoms is.

9.6.12 Basic rules for datasets, in which all proteins belong to one family

Often, protein families are defined as having 25% sequence identity, but other definitions may be appropriate as well. Families defined by a sequence homology threshold, may be suitable for many studies, especially when the identity among its members is fairly above this threshold (e.g. histones or many mammalian proteins). However, many interesting proteins fall into the *'twilight zone'* where sequence-based methods tend to fail, like proteasomal subunits or aminoacyl-tRNA-synthetases. Here, the fold databases will provide a more reliable classification. Also, they define domains in the sequences, which may differ between individual structures.

9.6.13 Basic rules for representative subsets of the PDB, having one interesting property

In general, folding classifications represent structural relationships much better than sequence clustering. In terms of coverage of the PDB, the SCOP database is advantageous of CATH, because it contains classification for many non-canonic folds like fibrous proteins, viruses, membrane domains, peptides, fragments etc. If one wants only typically globular proteins in the dataset, the CATH database is best suited to exclude the un-

wanted structures. Both databases contain four main levels of hierarchy, and the nodes on the second level are mostly not based on sequence homology (*fold* in SCOP, *architecture* in CATH). On the third level (*superfamily* in SCOP, *topology* in CATH), obvious sequence homology occurs. Thus, the superfamilies within a fold are unique in some respect, and the use of the SCOP superfamily or CATH topology as the basis for creating representative sets is strongly recommended. From these superfamilies, the structure with the highest rank in a PDB sequence cluster should be used (see A.2.10).

10 Packing analysis reveals functionally important regions in all datasets, where many cavities occur

10.1 Packing analysis of reference data

10.1.1 Packing densities in the reference dataset

The packing densities for all atoms in the 87 reference protein structures have been calculated using the improved Voronoi procedure (see section 8.3). Each atom was assigned to one out of 18 chemical atom types (see table 8.5) and one or several out of seven atomic subsets (explained in section 8.2). As shown in table 10.1, about one half (52.2%) of the protein atoms belong to the surface. About one third (36.6%) of the atoms were directly below the surface, and only the remaining 11.2% were deeply buried. Containing 18.1% of the atoms, the BOTTOM subset was larger than the DEEP set. About every 25th atom (3.8%) in the set was a direct neighbor of a cavity, and about 60% of the cavities are occupied by water. For analyzing packing density, only 1.6% of the neighboring atoms of empty cavities were considered. The three subsets, SURFACE, SHALLOW and DEEP, are clearly visible as three distinct concentric layers. Visual inspection revealed that many structures have more than one contiguous region, classified as DEEP, often relating to protein domains or subunits (for an example, see figure 10.1).

The average local packing densities for the 18 ProtOr atom types and each subset are shown in table 10.1. In all atomic subsets, the tertiary nitrogen atoms from arginine (atom type N3H0u) were packed most tightly, while reduced cysteine groups (S2H1u) were packed most fluffily in all subsets except SURFACE. The average packing densities range from 0.333 (peptide and amide oxygen O1H0u) to 0.758 (N3H0u) for surface atoms and from 0.493 (O2H1u in CAVNB) to 0.868 (N3H0u in BOTTOM). The standard deviations of the average packing densities were similar for all subsets (5-12%) except the surface atomic subset (8-31%). For a few average packing densities in the small atomic subsets CAVNB and DEEP, the deviation was higher due to a smaller sample size. Similar deviations were found earlier in analyses which reported reference packing values [Pontius et al., 1996, Tsai and Gerstein, 2002].

To test the reliability and uniformity of the dataset, the local packing of all atoms in the BURIED subset was compared to the average packing densities given in table 10.1. The packing rms Z_{rms}, defined in equation 8.2, was calculated for each structure using the average packing for the BURIED subset as reference values. For the set of 87 high-resolution structures, the Z_{rms} was within the range of 0.74-1.26, having an average of 0.96 ± 0.10.

The packing densities of each atom type in an arbitrary subset depend strongly on the hybridization state of the heavy atom and the number of hydrogens. Tertiary carbon (C3H0s,b) and nitrogen (N3H0u) atoms in the sp^2-state are packed most efficiently, while methyl groups (C4H3u) and the H-bond forming hydroxyl- (O2H1u) and thiol (S2H1u) groups exhibit the lowest packing density. Thus, average packing densities for different atom types indicate how these atoms are covalently linked. Comparing the atomic packing density of the atom types in different atomic subsets revealed that most differences are fairly below the standard deviation. Atoms close to internal cavities are packed less efficiently. The lowest packing densities are observed for atom types having few covalently linked atoms.

In general, the BOTTOM subset has the highest average packing densities of all

subset	BURIED	BURIED	SHALLOW	DEEP	CAVNB	LIGNB	BOTTOM	SURFACE
	percentage	47.7%	36.5%	11.2%	1.6%	-	18.1%	52.3%
atom type	N(atoms)	63734	48795	14939	2121	7520	24156	69747
C3H0b	4298	0.803	0.803	0.804	0.701	0.834	0.815	0.608
C3H0s	11710	0.818	0.819	0.811	**0.727**	0.865	0.824	0.718
C3H1b	2342	0.610	0.618	0.599	0.539	0.646	0.616	0.449
C3H1s	1988	0.635	0.638	0.624	0.568	0.650	0.639	0.451
C4H1b	4768	0.753	0.755	0.747	0.658	0.788	0.759	0.571
C4H1s	6389	0.784	0.784	0.783	0.695	0.807	0.789	0.607
C4H2b	1255	0.649	0.653	0.643	0.581	0.675	0.654	0.410
C4H2s	4756	0.664	0.665	0.658	0.595	0.681	0.670	0.459
C4H3u	4387	0.577	0.584	0.570	0.530	0.606	0.588	0.409
N3H0u	569	**0.853**	**0.852**	**0.864**	-	0.868	**0.868**	**0.758**
N3H1b	543	0.684	0.690	0.663	0.554	0.703	0.684	0.430
N3H1s	10330	0.754	0.755	0.745	0.664	0.759	0.758	0.578
N3H2u	351	0.612	0.617	0.602	0.540	0.586	0.613	0.322
N4H3u	19	0.673	0.682	0.622	-	0.515	0.671	0.266
O1H0u	8909	0.608	0.612	0.596	0.526	0.605	0.607	**0.333**
O2H1u	746	0.595	0.604	0.581	**0.493**	0.593	0.599	0.318
S2H0u	297	0.616	0.627	0.599	0.544	**0.686**	0.623	0.458
S2H1u	74	**0.570**	**0.580**	**0.546**	0.513	-	**0.581**	0.445

Table 10.1: Average packing densities for 18 atom types of the ProtOr set. The atom types are listed in the Format ExHyZ, where E is the element, x the number of covalently linked atoms, y the number of bonded hydrogens and Z a letter for further distinction. The packing density is calculated as $PD = \frac{V_{vdW}}{V_{total}}$. Average packing densities for five atomic subsets built from 87 high-resolution structures and for the LIGNB subset of the hetero group dataset are given. For the first subset, the number of atoms is also given. The minimum and maximum packing densities for each subset are highlighted.

atomic subsets, followed by the SHALLOW subset. By its definition, it seems self-evident that the BOTTOM set encloses the most deeply buried protein atoms. Surprisingly, however, a lower packing efficiency was observed in the DEEP subset compared to the BOTTOM subset. To interpret this surprising finding, it was investigated whether the DEEP and BOTTOM subsets were at all in the same region. For this, it was counted how many atoms in them were identical in each protein structure. The overlap between the BOTTOM and DEEP subsets is widespread: 72±24% of the DEEP subset's atoms of one protein are also in the BOTTOM subset. Extending this analysis to cavity neighbors, it was found that 27 ± 26% of the CAVNB subset's atoms of one protein are also in the DEEP subset, but only 3 ± 7% of them also belong to the BOTTOM set. This data shows clearly, that many cavity neighbors are removed from the BOTTOM but not from the DEEP atomic subset. An example of this effect is shown in figure 10.1. In the following section, it will be elucidated, where the cavities themselves reside.

10.1.2 Internal cavities in the reference dataset

Using a 1.4Å radius probe, cavities were found in all protein structures larger than 150 amino acid residues. 522 of 962 cavities contained at least one water molecule, still providing enough empty space for the probe. A range of 0.0 to 9.8 cavities per 100 residues with an average of 4.4 cavities per 100 residues was found in the reference dataset.

The 962 cavities were surrounded by 6,888 atoms. 1,576 of them belong to the SURFACE atomic subset, 3,014 to the SHALLOW and 2,298 to the DEEP subset, suggesting that cavities occur more frequently in the deep interior, or at least at its border. These cavity neighboring atoms have virtually the same packing densities as the CAVNB atomic subset, in which only buried neighbor atoms of non-occupied cavities were included.

In each subset from SURFACE to DEEP, the percentage of nonpolar atoms was calculated for both cavity neighboring atoms and atoms in the entire subset, as shown in figure 10.3. Cavities closer to the hydrophobic core, have slightly less polar neighbors. Their ratio increases from the surface to the deep interior, from 1.29 to 1.71. This trend is independent of the polarity of the entire subsets, as the SHALLOW atomic subset contains less nonpolar atoms than the other subsets. A majority of 84% of the cavities has at least one polar neighbor atom. In the protein core, only 78% of the cavities have one or more polar neighbor atoms.

To get information, in which depth from the protein surface cavities preferably occur, the distance to superficial atoms was analyzed at a higher resolution. The surface distance *sdist* for all buried atoms was collected and assigned to 20 shells, each having an equal number of atoms, 2,957±1 atoms. The occurrence of cavity positions and cavity neighboring atoms in each shell is shown for proteins larger than 100 amino acid residues in figure 10.2. Cavities occur most frequently in the shells 12-14 (2.50- 3.60Å) (figure 10.2 a)). A small second increase is found in shell 20 (below 6.10Å). Between shells 15-19, fewer than half as many cavities per shell were counted than in shell 12. Very few cavity positions occur closer to the surface than shell number 11 (2.42Å). In contrast, the neighboring atoms of cavities follow a more regular distribution (figure 10.2 b)). The percentage of cavity neighboring atoms among all atoms increases from outer to inner shells, reaching 12% for the innermost shell. Cavity neighboring atoms are also found in the outermost shells. When medium-sized proteins (100-200 aa) are treated seperately, the shells would not extend as deeply as in large proteins. Therefore, cavities are found

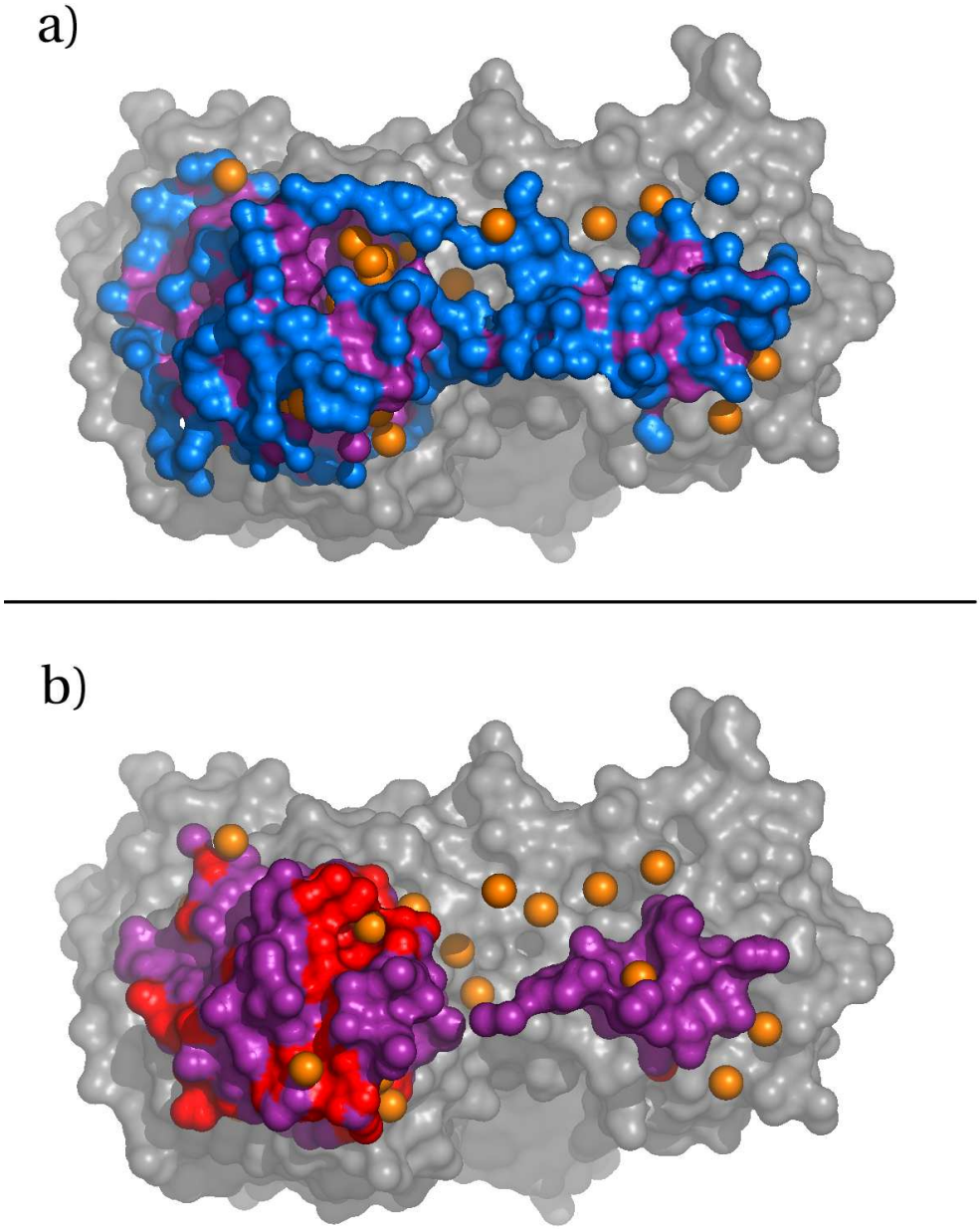

Figure 10.1: Alternate definition of the protein core displayed for the structure of Leu-Aminopeptidase (PDB-code: 1lcp). The grey background represents the protein envelope (SURFACE and SHALLOW atoms), while the protein core atoms are highlighted according to two different atomic subset definitions: a) The BOTTOM set defined by Tsai [Tsai and Gerstein, 2002] (blue). b) The DEEP subset from this study viewed from the same angle (red). Purple atoms illustrate assignment to both the BOTTOM and DEEP set. Cavity positions are indicated by orange spheres. For a), two layers of surface atoms were stripped away. For b) the distance to the Connolly surface was used. The figures were created using the open-source molecular viewer PyMOL (*www.pymol.org*).

less frequently in the innermost shell.

The question becomes, how did the contradictory localization of cavities, mentioned in section 5.1, emerge? On one hand, it was found that no cavities occur in the interior, defined as the BOTTOM set, because of a high packing density [Tsai et al., 1999]. In the DEEP subset, the packing density is lower, because many atoms are in direct contact with cavities. When the BOTTOM set is created by removing the surface atoms, many cavities - most of which were found at that depth - become opened up to the solvent and all of their neighbor atoms are also not counted. Furthermore, packing defects smaller than 2.8Å in diameter may also contribute to this effect. Thus, the definition of the BOTTOM set selects the most tightly packed atoms.

On the other hand, Hubbard [Hubbard et al., 1994] described the localization of most cavities in the center of protein structures. In their study, a probe radius of 1.25Å was used on a dataset containing much smaller structures. Though this probe radius is useful for finding cavities in homologous proteins, many cavities near the surface become solvent-accessible. The 1.4Å radius was used for compatibility with the study by Tsai [Tsai and Gerstein, 2002].

In addition, cavity positions were analyzed by Hubbard [Hubbard et al., 1994] constructing ellipsoid bodies around the proteins. However, some protein structures have a very irregular shape, which is ignored by this procedure and may even produce an ellipsoid with its center near the protein surface (see figure 10.1 for comparison). These effects result in a higher concentration of cavities in the protein core than is reported here.

In case of internal cavities, the polarity of the surrounding atoms rises from 56% nonpolar atoms near the protein surface to 63% in the deep interior. Interestingly, the polarity of all atoms in the SHALLOW (44%) and BOTTOM (49%) subsets is lower than in the remaining subsets. This is mainly due to the distribution of protein backbone atoms: peptide oxygen atoms are found more often in the SURFACE subset, while peptide nitrogen and carbon are found mostly in the SHALLOW subset (data not shown). It was found that nonpolar cavity neighbor atoms are, in general, more frequent than nonpolar atoms in the corresponding atomic subset (see figure 10.3). However, nonpolar atoms do not generally need to be packed less efficiently in the protein interior, because cavity neighboring atoms are only a small proportion of it. In fact, the data in table 10.1 reveals no remarkable differences. Earlier reports of aliphatic atoms being packed somewhat more tightly [Harpaz et al., 1994] are hard to compare to these values, due to the different atom radii used.

The observed packing in the deep interior is supported by molecular dynamics simulations done by Kocher, revealing that cavities are more likely to occur in nonpolar regions than in a polar environment [Kocher et al., 1996]. One reason given for this was that nonpolar regions often have greater flexibility, because sidechain atoms are less constrained by polar interactions and hydrogen bonds. Also, by containing less polar atoms at the surface of cavities, a higher number of polar interactions is possible in the protein interior, contributing to protein stability.

There is in general, heterogeneity in packing: In most analyses, the standard deviation of reference packing values is roughly 5-12%. In fact, by analyzing the scaling of geometric properties, such as volume and number of cavities with protein size, packing in the protein interior was found to be comparable to a collection of random spheres; packing in large

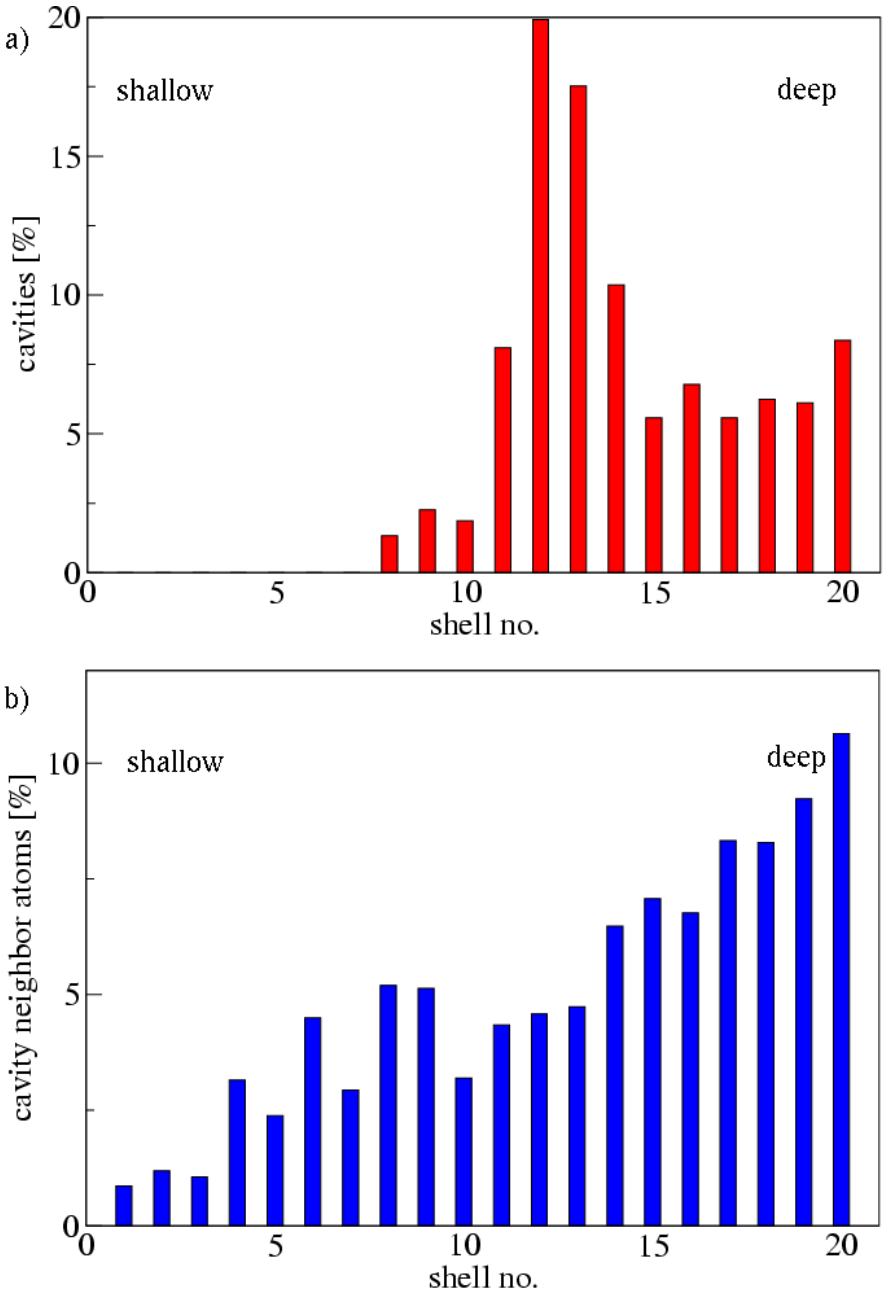

Figure 10.2: The distribution of a) cavity positions and b) cavity neighbor atoms in different depths from the surface is shown for proteins with at least 100 amino acid residues. All protein atoms were distributed among 20 concentric shells, according to their distance to the next surface atom (*sdist*). Each column shows the percentage of all cavity positions/cavity neighbor atoms found in that shell. The distance ranges for the shells were chosen, such that all shells have the same number of atoms.

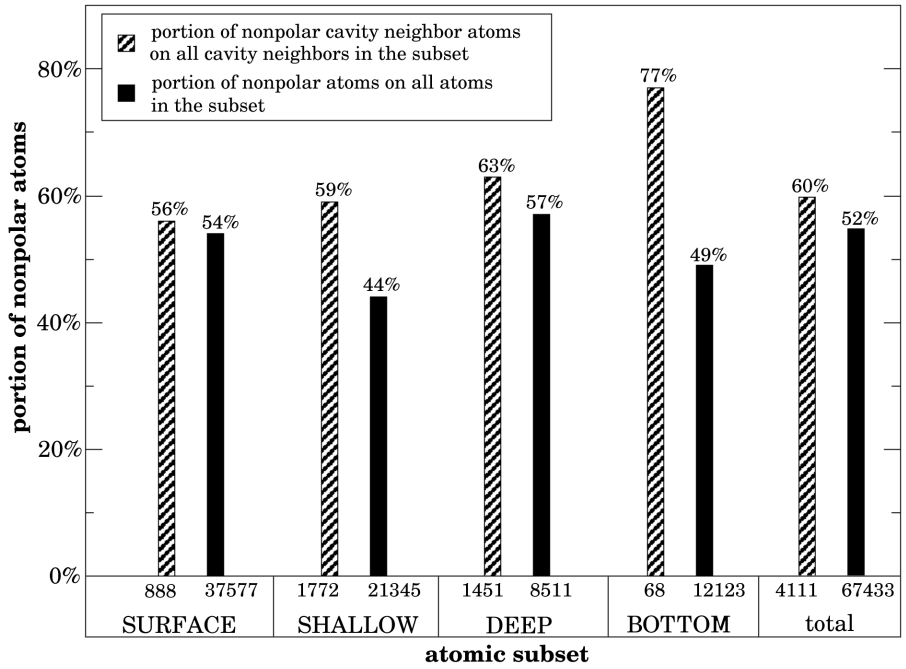

Figure 10.3: Polarity of cavity neighbors and other atoms in different atomic subsets. Two groups of bars indicate the relative amount of nonpolar atoms (all but nitrogen, oxygen and peptide and amide carbon atoms). These carbon atoms have a partial charge similar to most oxygen and nitrogen atoms due to the strong electronegativity of their neighboring atoms [Brooks et al., 1983]. The full bars illustrate polarity for all atoms of the particular subset, showing an increase of polarity for the tightly packed SHALLOW and BOTTOM subsets. Striped columns illustrate, how many nonpolar cavity neighboring atoms (CAVNB) are found among all cavity neighbors in a particular subset. The percentage of nonpolar atoms is above, their total number below each bar.

structures is similar to a model liquid, due to the occurrence of cavities [Liang and Dill, 2001]. It was also reported that about two thirds of all residues approximately coordinate themselves in face-centered cubic packing, while the remaining third does not [Bagci et al., 2003].

The data shows that packing in the protein interior is not entirely random: The largest number of cavities is seen at a depth of roughly 3.0Å. It would be the result of the polypeptide chain with its structural constraints folding to a globular shape. By using sequential Monte Carlo simulations the scaling of packing density with chain length, it was found likely to be a generic property of compact polymers [Zhang et al., 2004]. The report suggests that protein packing has not been optimized much by evolution. Packing is similar in all protein structures, depending on their size.

10.2 Packing analysis of transmembrane domains

10.2.1 Packing densities in the transmembrane dataset

A more focused analysis was carried out on 20 structures of helical transmembrane domains. These have been divided into *gates* and *coils* according to their function. A set of helical domains of globular proteins was used for comparison. Average packing densities were calculated from all buried atoms participating in helix-helix contacts in the transmembrane parts of the helices and the reference helices.

The size and atomic composition of a helix-helix contact patch varies with the chosen cut-off. The average packing densities for seven cut-off values were compared (see figure 10.4). All seven cut-off values produce qualitatively the same result. With a low cut-off the differences in packing between the three datasets are most drastic, but the total number of atoms is low. For a high cut-off value, the situation is inverted. As a compromise, all packing values were calculated at a mean cut-off value of 1.5Å.

The data in figure 10.5 shows that the helical interfaces of membrane coils are packed as tightly as in globular proteins (average packing density value $< PD >= 0.81$). By contrast, transmembrane helices of membrane channels and transporters are significantly (t-test, p-value of 0.005) less well packed with an average packing value of $< PD >= 0.79$.

It was found that virtually all amino acids contribute to the lower packing density of the membrane gates. Only Ala, Gly, and Ile are packed as well in the helical contacts of membrane coils and globular proteins. By sorting the amino acids according to their water solubility, it becomes apparent that the polar amino acids contribute most to the observed packing deficiency. These residues are again more abundant in the helix-helix interfaces of membrane gates, whereas the aromatic residues (Phe, Trp, and Tyr), which encompass the highest packing values in all datasets, are observed less frequently. These proportions approximately resemble those of the entire transmembrane helices [Hildebrand et al., 2004]. This observation justifies the method of concentrating on the interface atoms. In addition, it seems that the lower packing density of membrane gates is already partly encoded in the primary structure.

The comparison of the atomic packing density values reveals further details: The side-chain atoms buried between helices ($< PD >= 0.77$) are significantly (t-test, p-value of 0.005) less efficiently packed in the dataset of membrane gates than the main-chain atoms ($< PD >= 0.80$). This is especially valid for the polar side-chain atoms ($< PD >= 0.64$). This difference is much smaller in the datasets of membrane coils and helices of globular proteins. When a higher cut-off value for defining interfaces was chosen, the helix patches

expand in all three dimensions and the fraction of main-chain atoms at the interfaces increases Because the backbone atoms are packed more efficiently, the packing density of the investigated helix-helix interfaces rises accordingly. This increase is most obvious in the dataset of membrane gates, because the side-chain atoms are packed most loosely here.

This approach yields a characteristic trait of helix-helix packing of membrane gates that has not been described previously: Compared to helix-helix interfaces in membrane coils and globular proteins, the amount of main-chain atoms is higher in the dataset of membrane gates. At a cut-off value of 1.5Å, nearly one-half of the atoms engaged in helix-helix packing of membrane gates are part of the main chain.

In the membrane coils and globular domains, only one-third of those atoms are main chain. This implies that helices in membrane gates tend to be closer to each other, which allows two adjacent helices to form a direct contact between their backbones. In contrast, helices of membrane coils and globular proteins are further apart and contact with nearby helices is formed by the side chains instead of the backbone.

It has been mentioned that the tight backbone-to-backbone packing is important for the stabilization of the tertiary structure of membrane proteins. This structural signature was successfully applied to the prediction of their tertiary structure [Fleishman and Ben-Tal, 2002, Liu et al., 2004a]. Accordingly, the C_α to O hydrogen bond is a strong determinant of stability and specificity in transmembrane helix interactions [Senes et al., 2001]. Interestingly, multiple hydrogen bonds of this type are predominantly

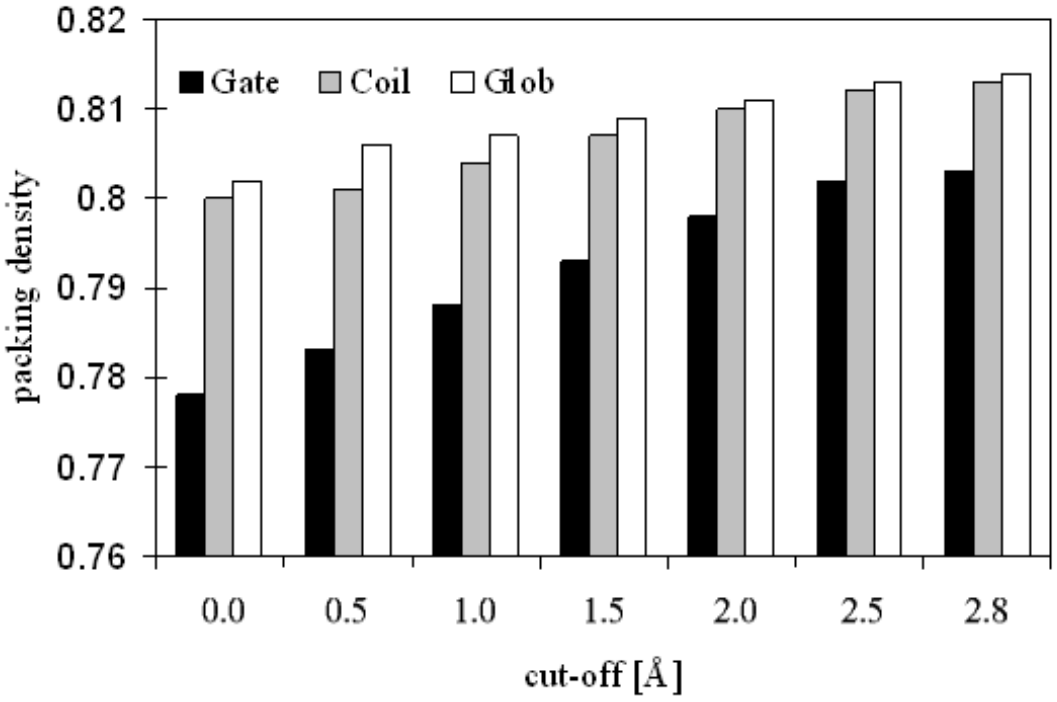

Figure 10.4: Comparison of the average packing densities $< PD >$ (columns) of all buried atoms in different datasets of membrane (Gate, Coil) and globular proteins (Glob) depending on the proposed cut-off values.

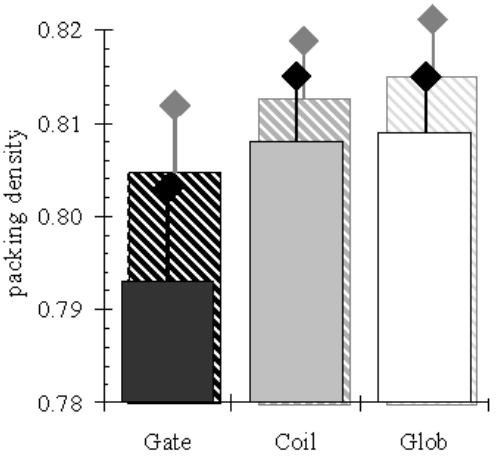

Figure 10.5: The average packing density (filled columns) of all buried atoms in different datasets of membrane (Gate, Coil) and globular proteins (Glob) are depicted with the according variances (black lines) of the collected data. When the atoms with direct contact to cavities are removed, the PD values increase differently as indicated by the columns in dashed lines (variances now in shaded lines).

found between parallel transmembrane helices that cross at right-handed angles. In this regard, membrane gates again differ markedly from the remaining helical membrane proteins: Two-thirds of the helix crossings are right-handed, whereas only one-third of the transmembrane helices in membrane coils cross right-handed [Hildebrand et al., 2004]).

These results suggest that helices, in membrane channels and transporters, are stabilized through close interactions of their backbones, which again facilitates the formation of multiple interhelical hydrogen bonds, stabilizing the topology of the helices. The helical mobility, which is necessary for the proper functioning of membrane channels and transporters [Swartz, 2004] could again be realized by the loose packing of their side-chain atoms.

In the following section, the question has to be addressed, whether the observed packing deficiencies found in the datasets are equally dispersed over the entire protein structure, or whether they cluster in certain protein regions, such as internal cavities.

10.2.2 Internal cavities in the transmembrane dataset

Transmembrane domains of membrane coils resemble nearly the same content of internal cavities as helices of globular proteins. In membrane coils, 5.2% of the buried atoms are in direct contact with a cavity, compared with 5.5% of buried atoms in globular helices (which is much more than the overall measure given in table 10.1). By contrast, slightly more cavities were found in the dataset of membrane gates (6.4%). Most cavities are either neighbored by 15-25 atoms (31%) or < 5 (57%) atoms. The results, therefore, indicate that the size of cavities found in proteins depends neither on the surrounding milieu, nor on the fold of the proteins [Rother et al., 2003].

Cavities in membrane gates are highly polar, in contrast to cavities in membrane

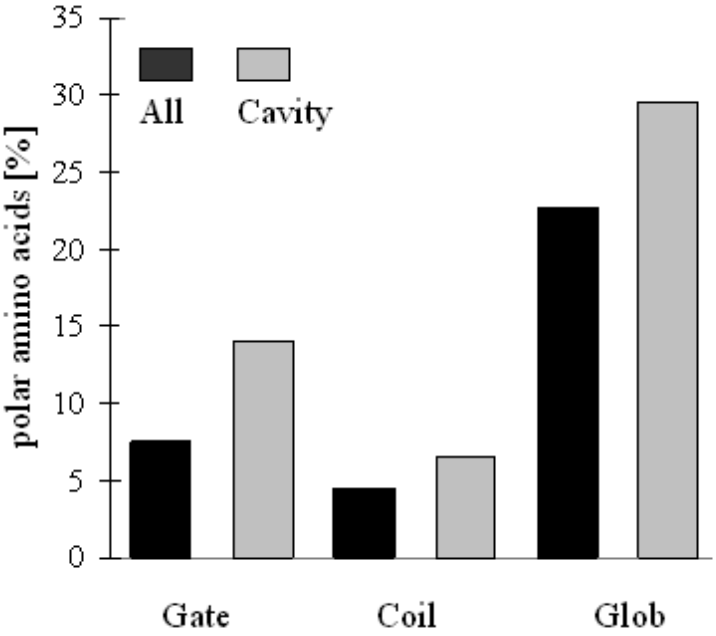

Figure 10.6: Portions of polar amino acids at interfaces between helices and surrounding cavities of membrane channels and transporters (Gate), other membrane proteins (Coil) and globular proteins (Glob).

coils (see figure 10.6). As shown above, most polar cavities in the reference dataset were found close to the protein surface. In contrast, transmembrane helices are encircled by hydrophobic lipid tails. Thus, polar cavities within membrane proteins are positioned close to those protein regions that communicate with the extra- and intracellular environments, as the pores that pervade these proteins (see figure 10.7). Therefore, it seems feasible that the cavities of membrane gates are encircled by more polar amino acids than the cavities of membrane coils and that many of them might be filled with water molecules.

On the other hand, it is likely that many of the polar cavities that are placed within the structures of membrane gates will surface when, the pores are opened via molecular rearrangements, and the residues previously surrounding the cavities become solvent-accessible. Therefore, it is not surprising that more cavities were found in those membrane gates that had been crystallized in a closed or partly closed conformation and that these structures (1iwg, 1pw4) are among those having the lowest packing density values. However, the packing density is influenced by the presence or absence of water in cavities and grooves at the protein surface [Tsai et al., 1999]. Thus, the cavities account for the reduced packing of membrane gates. The density distribution of membrane gates is, therefore, reflected by the localization and characterization of internal cavities.

Ongoing, nonpolar cavities in protein regions have been detected that are presumed to be functionally important as well. In the translocon (1rh5), these cavities are located around the restriction region and the so called plug. In the structure of the glycerol-3-

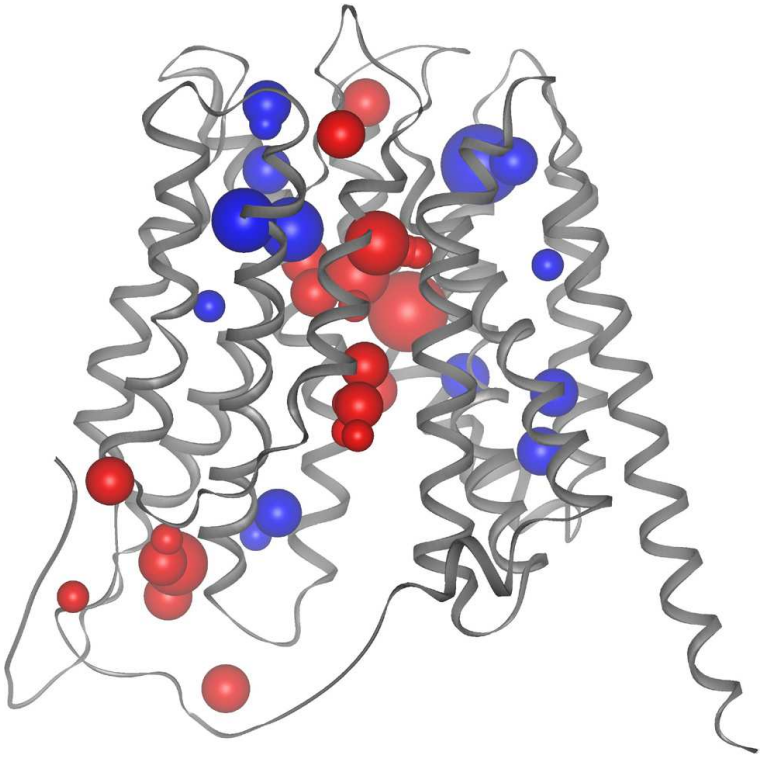

Figure 10.7: Polar (red) cavities are positioned predominantly in helix cap regions that are exposed to the polar milieu or within the gated pore of the glycerol-3-phosphate transporter (PDB code 1pw4). Nonpolar (blue) cavities are placed in the proposed hinge regions that facilitate the rocker-switch type movement of the helices that occurs upon substrate binding. The cavities are depicted as balls that are sized according to the number of atomic neighbors. The centers of the cavities were calculated from the atom coordinates of the cavities neighbor atoms.

phosphate transporter (1pw4), they are around the hinge regions (see figure 10.7). It is concluded that local packing defects allow for the structural flexibility that is required for the proper functioning of membrane channels and transporters.

Nevertheless, if the proper functioning of membrane channels and transporters is based on a relatively loose packing of helical interfaces? How are these proteins stabilized when compared to other membrane proteins? The close packing of the transmembrane backbones indicates that main-chain interactions probably compensate for the loose packing of side chains. The structural details investigated above promote a better understanding of the relation between stabilization and function of membrane proteins.

10.3 Packing analysis of protein-bound ligands and coenzymes

10.3.1 Packing densities of protein heavy atoms in binding sites for ligands and coenzymes

To characterise packing in binding regions for organic compounds, two distinct questions were adressed: Is the packing density of protein atoms surrounding a ligand similar to normal buried atoms? Do the ligands have a packing density comparable to that of proteins? To answer these questions, a total of 7,520 atoms in the LIGNB atomic subset and 13,717 hetero group atoms were analyzed.

From the data in table 10.1, it is clearly visible, that many neighboring atoms of hetero groups are packed more tightly than other buried atoms. Only for the atom groups N3H2, N4H3, O1H0 and O2H1, which frequently participate in hydrogen bonds, can lower average packing values be observed. Sulfur atoms (S2H0) were most distinct, for which $< PD_{loc} >$ rises by 0.07 compared to the BURIED atomic subset. The latter observation can be explained by many of the 40 ligand-binding sulfur atoms from cysteine and methionine residues participating in the covalent binding of porphyrine groups. Compared to a disulfide bond, the corresponding sulfur-carbon bond is much shorter. This gives a higher packing density despite the lower radius of the carbon atom.

Another observation is that the proportion of positively charged amino groups (N4H3u) is significantly increased (1.2% in the LIGNB atomic subset, instead of 0.03% in the BURIED subset). This can be explained by the positive charges, being actively involved in binding negatively charged and polar groups. A similar effect can be expected for the negatively charged carboxy groups, but they belong to the same ProtOr atom type as the backbone oxygen (O2H0u), which blurs any effect.

Obviously, the majority of protein ligand neighbors are packed more tightly than atoms in the protein interior. Two reasons could account for this: Either the protein atoms participating in ligand binding are very close to the ligand atoms, or they just pack more tightly among each other. Nontheless, a very high proportion of cavities is found there: among 1kDa of ligand neighbors, an average of 9.2 cavities is found. This does not contradict a high packing density, as already observed in the reference data. In the following section, this shall be elucidated by considering the packing of hetero groups.

10.3.2 Packing densities in small organic molecules and coenzymes

Out of 31 hetero groups in the initial dataset, 13 provided enough buried atoms to allow for a reasonable calculation. For these six coenzymes and seven small organic compounds, packing densities were calculated.

Because the hetero group atoms differ chemically from amino acids, it was unclear

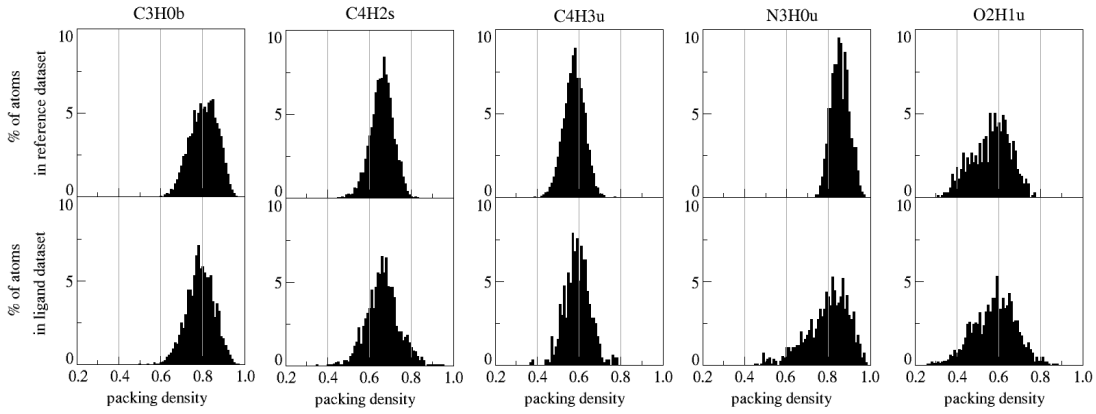

Figure 10.8: Histographic distribution of packing densities for five ProtOr atom types. In the top row, the packing data from 87 reference proteins were used, in the bottom row data from 1333 structures containing 6 types of coenzymes and 7 different small organic molecules.

whether they were at all comparable to the packing data presented above. For this reason, the histographic distribution of packing densities for some atom types was compared between hetero groups and reference data. In figure 10.8, it becomes clear that all but one curve are very similar in both sets. Because of the lower numbers, the histograms for hetero groups appear a little more noisy. The distribution for the N3H0 atom type (aromatic and tertiary amide nitrogen), is much more spread out and contains many atoms with low packing densities. It was discovered that some of the nitrogens (namely AN7 and AN3) in the NAD hetero group are responsible for this effect.

The shape of these curves suggested a test for normality. On all 18 ProtOr atom types, in both reference and hetero group data, the Kolmogorov-Smirnov-test was applied ($p < 0.05$). All hypotheses that the data for a particular type was normally distributed, were rejected. This analysis was expanded to all chemically distinct positions in each ligand (e.g. for the carboxyl-carbon in acetate ACT). All but three tests were negative, but given the high number of tests (246) even these three are not of much significance, since they correspond to the number of tests one would expect to give false positives with $p < 0.05$.

For these reasons, the packing data was analyzed independently for all chemically distinct atoms (subsequently called chemical atom positions), even if they had the same ProtOr atom type. The number of atoms in each position was distributed unevenly and, for some of them, very small. Thus, the data for chemical atom positions having less than 20 atoms were not used. This indicates that some parts of the protein-bound hetero groups are preferably oriented towards the protein surface, while other parts preferably point to the bottom of the binding pocket.

The packing densities were visualized, in the form of boxplots for each ProtOr atom type, to compare them to the reference dataset (see figures 10.9 and 10.10 for an example). To inspect a particular hetero group in detail, the average packing density of each chemical atom position was mapped on a sample 3D-model. Figure 10.11, 10.12 and 10.13 show both the absolute values of $< PD >$ and the deviation from reference data for the

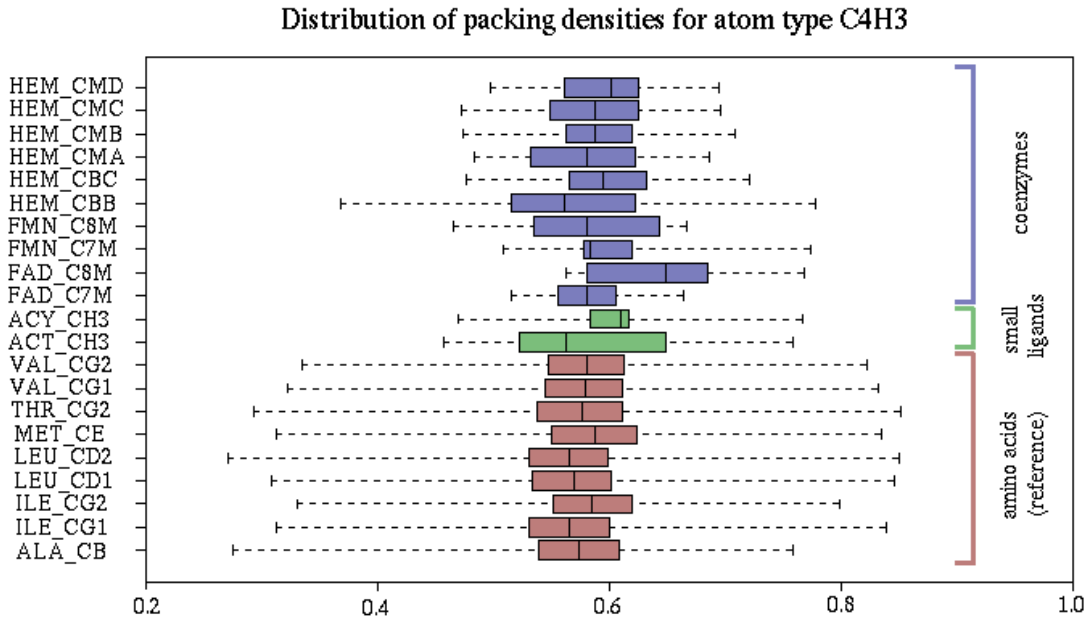

Figure 10.9: Packing densities of methyl carbons (ProtOr type C4H3). Each of the boxes show the two medium quartils within the data, while the whiskers display the extreme values. In the plots, the blue bars indicate packing of coenzymes; the green ones are for small organic molecules; and the red ones are based on packing data from the reference dataset.

corresponding ProtOr type represented by a color scale.

10.3.2.1 Packing densities in small organic compounds

Of the initial 16 small molecules, only seven provided enough data for statistical analysis: acetate (ACT/ACY), β-mercaptoethanol (BME), ethylene glycol (EDO/EGL), formic acid (FMT) and glycerol (GOL). The others only had buried atoms in a few structures or got excluded before the calculation because only few non-redundant structures existed for them. The atom numbers for each position ranged from 30-60 for most positions, except for EDO and GOL, of which they were significantly higher (90+), and for some atoms in BME (less than 30). Because of this, the standard deviations of the average packing densities are in many cases, slightly higher than in the reference data.

Several aliphatic carbons in acetate (ACT and ACY) (see figure 10.10), mercaptoethanol (BME) and glycerol (GOL) had higher average packing densities than the reference. The remaining chemical atom positions were discovered to be similar to the reference data: In figure 10.10, the green bars at first appear to be badly packed oxygen atoms from carboxyl groups. But they are, in fact, similar to the carboxyl groups from amino acids (the lines GLU_OE1, GLU_OE2, ASP_OD1 and ASP_OD2 in figure 10.10), while the other red bars are carbonyl groups. The same applies to the carboxyl carbon atoms in acetate. This shows that the packing densities of small organic compounds are, in average, as high or even higher than those of chemically equivalent atoms in amino acids.

The relative amount of cavity neighbor atoms is increased in all hetero groups studied.

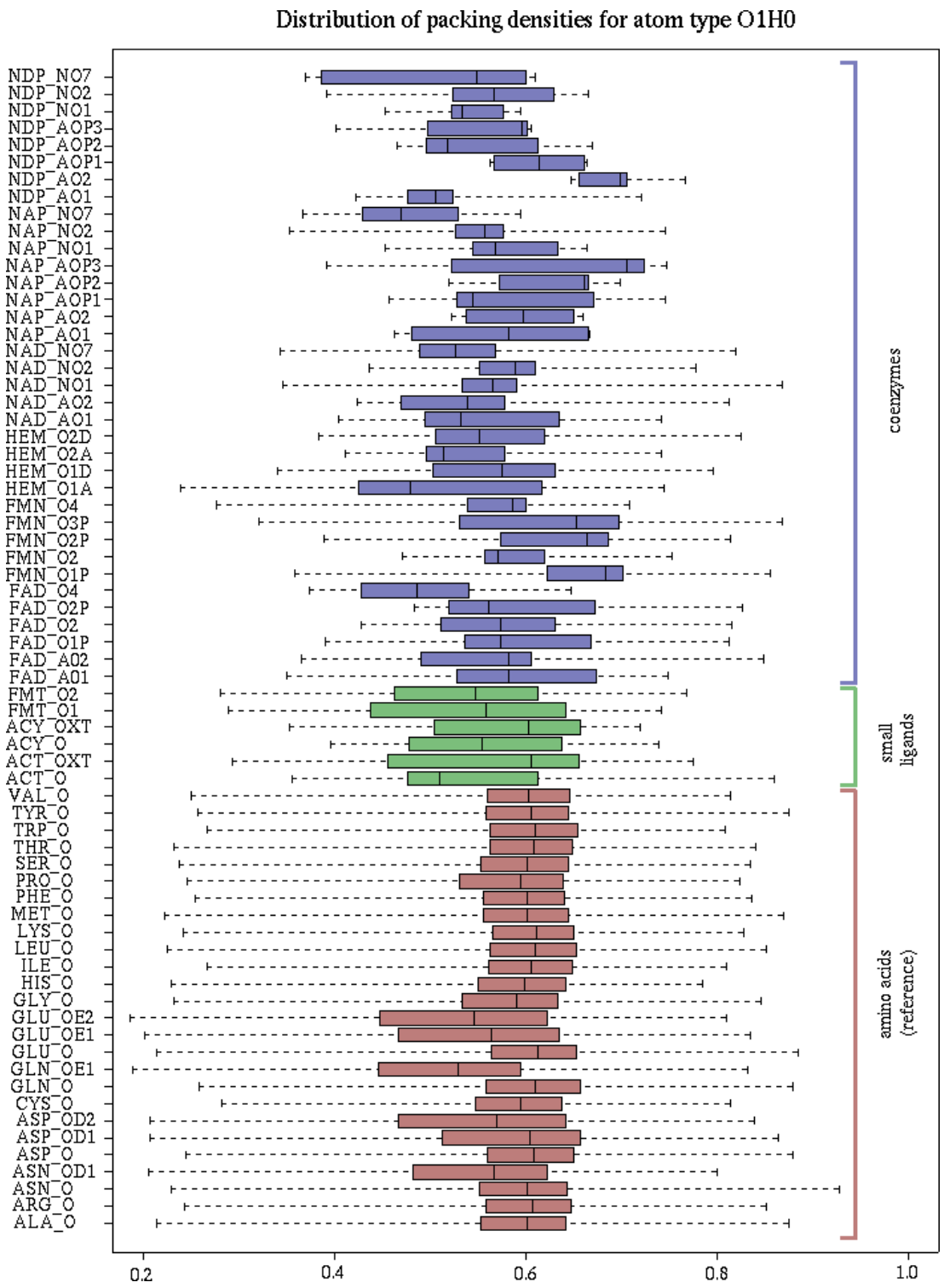

Figure 10.10: Packing densities of carbonyl and carboxyl oxygens (ProtOr type O1H0). Each of the boxes show the two medium quartils within the data, while the whiskers display the extreme values. In the plots, the blue bars indicate packing of coenzymes; the green ones are for small organic molecules; and the red ones are based on packing data from the reference dataset.

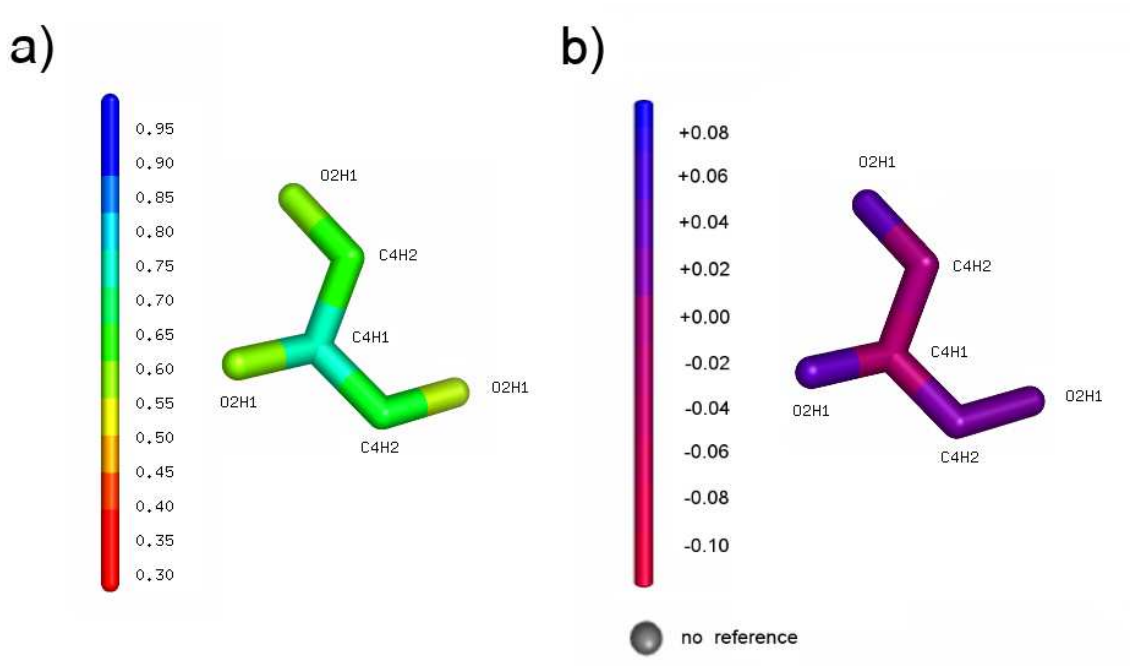

Figure 10.11: a) Average packing densities of glycerol (GOL) (red 0.2, blue 0.9). b) Packing density relative to the reference.(blue - tighter packing than reference, red - worse packing than reference). The pictures have been created with PyMOL (*www.pymol.org*).

Compared to the reference data, it is higher by around one order of magnitude. Between about 10% and 35% of the atoms are neighbors to at least one atom-sized cavity. In the most extreme case, one of the oxygens in formic acid, 41% cavity neighbors were found. In average, there are 6.9 cavities per 1kDa ligand atoms, compared to 9.2 cavities per kDa ligand neighbor atoms. This means that cavities occur about one order of magnitude more frequently in the vicinity of ligands than in the reference dataset. Despite the high amount of cavities, the packing densities are still similar to the reference data.

Of course, most of the cavities are occupied by crystallographically resolved water molecules, by which they mediate the interactions between a ligand and its binding pocket. This is supported by the observation that cavities are preferably located near atoms capable of participating in hydrogen bonding, such as O1H0 (see figure 10.10). In the native state, even those cavities can be expected to be occupied by solvent, for which no exact position could be determined. Therefore, the actual packing of ligands is even more tight than the average packing densities indicate. This conlusion is strongly supported by the tight packing of the LIGNB subset.

10.3.2.2 Packing densities in coenzymes

Of the six coenzyme types considered, the atom numbers were generally a little higher (around 50 heavy atoms per position) than for the small ligands (mostly 20-30). Many observations made of the small molecules still hold: Many average packing densities are almost the same as those of chemically similar reference types, and some average packing densities are even higher. There is a tenfold increase in the number of cavity neighbors compared to protein atoms. Most standard deviations are increased. This increase was

even larger for oxygen atoms that can participate in hydrogen bonds. Atoms that actually have a hydrogen bond were not distinguished from those having none. In the future, this should be considered in order to assign both types different atom radii.

An important distinction was observed for atoms at the borders of the molecules. Most groups protruding from a coenzyme, such as methyl groups in heme (HEM, see figure 10.13), mehoxy groups in flavin (FAD, FMN, see figure 10.12), and hydroxyl groups in flavin and nicotinamide nucleotides (NAD, NAP, NDP), were packed very tightly, while the atoms in aromatic ring systems in the middle were packed equally or less well-packed as the reference. This gives the tight interaction between ligands and binding pockets further support.

There were some nitrogen chemical atom positions, particularly in the adenine rings of nicotinamide nucleotides (NAD, NAP and NDP), that were packed much less tightly than the reference. It also has to be noted that the nitrogens in the tetrapyrrol system of heme groups (of the N3H0 type) also show a lower packing density. This indicates that the complex bond is not represented very well by the N3H0 type. For some chemical atom positions, like some aromatic nitrogens (N2H0) and ester oxygens (O2H0), no suitable reference was found among the protein atoms. In the flavins (FAD and FMN) this is annoying, because two of the N2H0 atoms are those binding the hydrogens.

It was tested whether the average packing densities of ligands/coenzymes and protein atoms can be described by a similar distribution. As the normality tests did not hold, one needs to use the Mann-Whitney (or Wilcox) test, instead of the t-test, to compare two mean values. For 241 out of 246 chemical atom positions, the mean values were not the same ($p < 0.05$).

A comparable analysis of packing densities was made on structures of nucleic acids [Voss and Gerstein, 2005]. Here, it was found that RNA is packed more tightly, with a standard deviation in the same range as in proteins. The picture for coenzymes and ligands is also similar. Owing to ligands being more heterogeneous, the data is not as consistent and more difficult to interpret as that on nucleic acids. The particularly loose packing and high standard deviation of oxygen atoms was also found for nucleic acids. In the study on RNA [Voss and Gerstein, 2005], an increased packing density was found most significantly for carbon atoms in aromatic rings, namely the atom types C3H0 and C3H1. In the coenzymes analyzed here, the opposite was observed. One suggested reason for that is the base pair stacking in nucleic acids, bringing atoms from those rings more closely together.

These results are in good accordance with the model describing binding of proteins to nucleic acids: Using the Columba database, a set of 224 structures for seven DNA-binding proteins was created. Both conformational changes of the protein backbone and of the local binding site were analyzed. It was found that the binding sites of all seven DNA-binding proteins follow the induced fit model. The backbone in five proteins occurs in a conformational ensemble, in which the DNA-binding conformation also occurs (published in [Günther et al., 2006]). In this part of the study, K. Rother was responsible for preparing the data and extracting binding site structures, while S. Günther did the 3D structural comparisons and statistical evaluation.

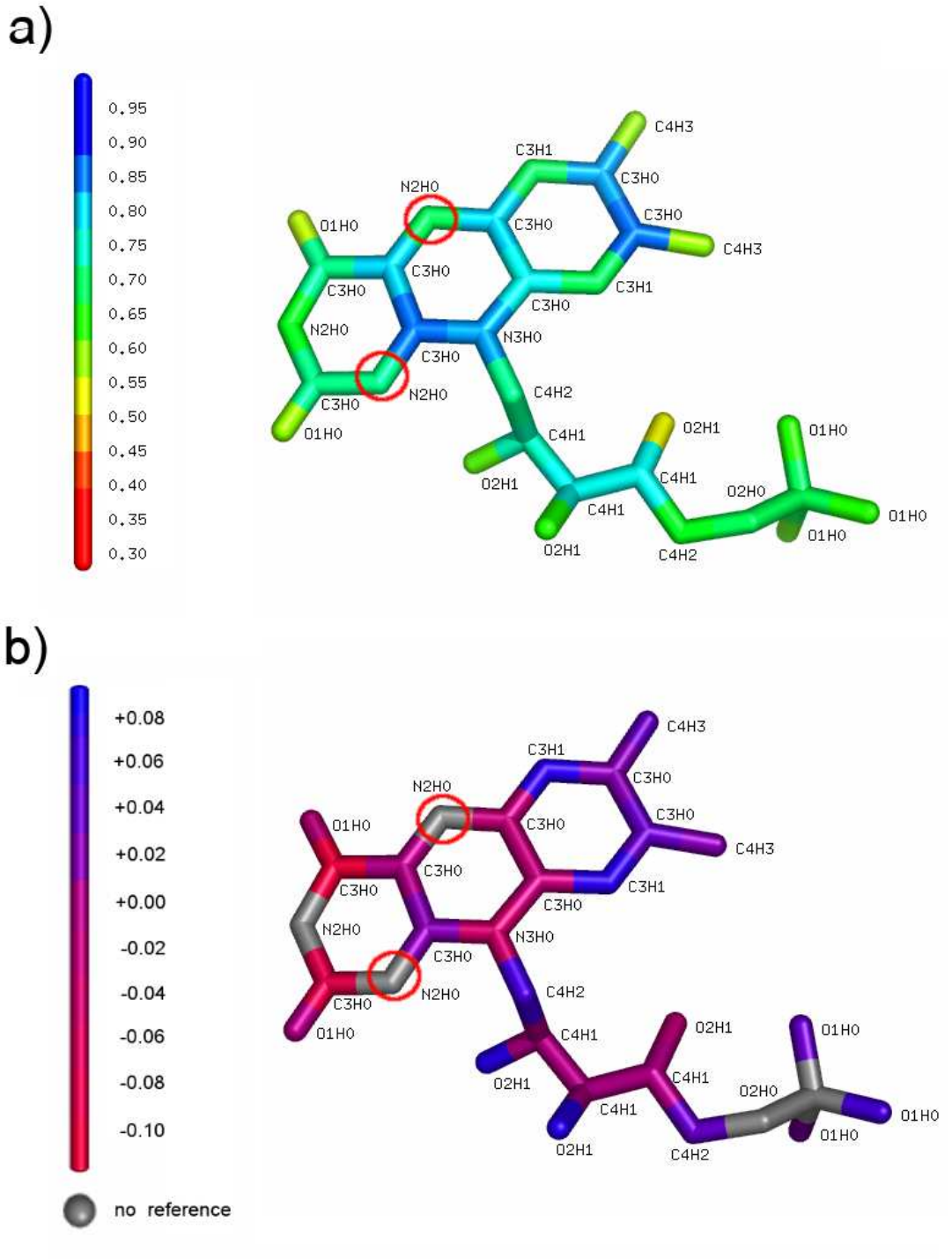

Figure 10.12: a) Average packing densities of flavin mononucleotide (FMN) (red 0.2, blue 0.9). b) Packing density relative to the reference (blue - tighter packing than reference, red - worse packing than reference). The two atom positions marked by red circles are the ones that can bind hydrogen. For the gray atoms, no reference data was available. The pictures have been created with PyMOL (*www.pymol.org*).

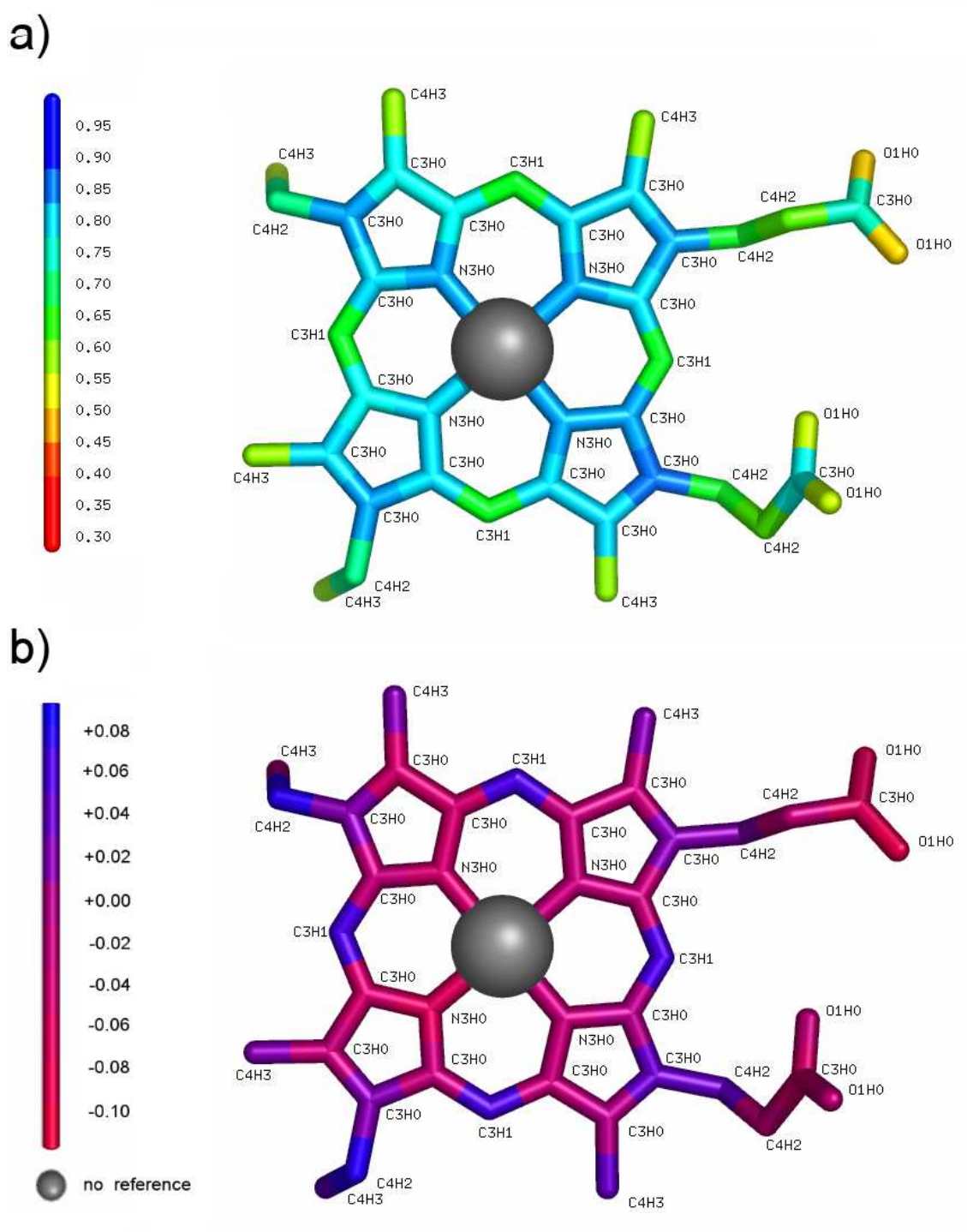

Figure 10.13: a) Average packing densities of protoporphyrin IX (HEM) (red 0.2, blue 0.9). b) Packing density relative to the reference.(blue - tighter packing than reference, red - worse packing than reference). The pictures have been created with PyMOL (*www.pymol.org*).

10.4 How can packing analyses be improved further?

The biggest drawback of the methods used here is that atoms at the surface of protein structures are not principally comparable to buried atoms. This is especially important in analyzing protein ligands, which, by definition, bind at the protein surface. In fact, this effect has greatly impeded the study of hetero groups, resulting in both a lower number of protein ligands that could be analyzed, and lowering the amount of data for the remaining ones.

An approach was considered to scale down the volume of surface atoms in such a way that they become comparable to buried atoms [Rother, 2002]. The solvent accessible surface area A_{SAS} could be used for the scaling. Using a linear or low-order polynomial scaling function like $V_0(A_{SAS}; V_{surf}) = V_{surf} - a * A_{SAS}$, pseudo-buried atomic volumes V_0 could be calculated from the atomic volume of a surface atom V_{surf}. The main problem with this method is that the resulting volumes would be very biased, since the variation of V_{surf} depends on A_{SAS}. Thus, the scaling function must take these different variations into account. A mathematically reasonable model is yet to be found.

The established methods for packing analysis (Voronoi-based, Alpha-shape and Occluded surface) are isotropic, i.e. the orientation of an individual atom does not affect the calculation. This ignores the fact that certain atom types prefer a specific chemical environment, and that this environment is anisotropic. An approach proposed by Rantanen [Rantanen et al., 2003] decomposes proteins into three-atom-fragments, thus defining the orientation of a central atom. Large numbers of the same type of fragments were superimposed together with the atoms surrounding each fragement. This revealed which types of neighbors of an atom were preferred in specific relative positions. These neighbors were grouped mathematically using the method of Gaussian Mixtures [Rantanen et al., 2001]. This procedure has been employed to characterize the environment of bound pyridoxal phosphate [Denesyuk et al., 2003] and adenine coenzymes [Denessiouk and Johnson, 2003] in extraordinary detail.

Another approach is to exploit information about an atom's mobility to characterise protein packing. Such information is represented in the b-factors of an X-ray structure. Also, data derived from molecular dynamics has been used to characterize cavities in a single structure [Kocher et al., 1996], and energy minimization were used for a similar purpose [Machicado et al., 2002]. However, none of these three methods was used for an extensive analysis of protein packing. Using any of these measures, it should be possible to improve packing calculations of both protein and ligand atoms in such a way that the chemical nature of each atom is represented in higher detail.

Part IV

Conclusions

11 The Columba database is a useful instrument to create protein structure datasets

The Columba database has been implemented as a relational database containing biologically relevant information on protein structures (published in [Rother et al., 2004]). Each structure from the PDB is annotated by secondary data, regarding protein-fold classification, enzymatic classification, participation in metabolic pathways, secondary structure, sequence data and others. In total, sixteen databases have been integrated into Columba. This approach, known as data warehousing, allows efficient cross-database surveys and data mining for customized protein structure datasets. Links, between these data and the protein structures, on the entry, compound, and chain level, are either taken from the second-party databases or are computed within Columba itself, leading to more accurate information than available in the PDB.

Quantifying the data in Columba, it was found that most of the data sources, which require manual annotation, lag behind the rapid growth of the PDB considerably. Roughly two thirds of all protein structures are well-annotated, while there is another poorly-annotated third. The latter includes not only recent structures, and nucleic acids, unfolded peptides, antibiotics and other small structures (published in [Rother et al., 2005]).

This observation leads to three conclusions important for data miners: First, the availability of cross-references to other databases should be considered in the creation of representative sets, because it is desirable to have datasets for which a high amount of secondary annotation is available. Second, the structures referenced by different sources are often the same, because different kinds of annotation sources overlap to a high degree. Third, researchers should be aware that choosing a set of structures, based on second-party annotation, may introduce a strong bias into the selection, because exotic structures like D-peptides, nucleic acids, etc. are suppressed. Most researchers will probably welcome, rather than be concerned by this effect. But unpleasantly, recent structures are also suppressed, *recent* stretching over a period of up to four years.

The lack of annotation for recently determined structures is worrisome. Since data is likely to accumulate faster than the budgets of databases relying on manual curation, the gap will become larger rather than smaller, although this claim cannot be proven. Unless methods for automated structure annotation are improving significantly, this *annotation gap* can only be closed by increasing the amount of resources invested into manual inspection and automatic annotation of protein structures.

The annotation provided by secondary databases on the same protein is often redundant. Rather than being a disadvantage, the redundancy provides an additional level of quality control: Data from multiple sources could be used to cross-check whether or not

the annotation of a particular protein is typical for its kind - possibly indicating interesting properties or annotation errors. Such a strategy, matching structural and sequence annotation for query improvement, has recently been proposed by the Biozon project [Birkland and Yona, 2006]. Designing queries for the creation of datasets often gives rise to a conflict between efficiency and completeness. Practically, one cannot strive for completeness without exhaustive manual inspection. By further growth of the databases, it will become more difficult to do this in a reasonable time frame. The seven questions from section 9.6.11, guiding the creation of datasets, are useful for developing advanced data mining strategies.

The Columba database was made available via a web interface that allows queries to be combined from multiple filtering conditions (published in [Trissl et al., 2005, Rother et al., 2006]. However, for complex tasks, a direct access to the database is necessary. Analogously, the interfaces of other integrated databases show this drawback. Even the highly sophisticated SRS interface fails to perform certain types of queries, e.g. *'get the entry with the highest sequence length from each family'*. As a consequence, the data should be accessible in a way that allows for highly customizable queries, thus avoiding the limits of web-form-based interfaces.

Data integrators should provide both convenient ways to access the data, and to make the entire data available. But for the latter point, a non-scientific concern comes into play: How can it be ensured that the scientists, who provide the original data, get merit (and funding) for their work flowing into an integrated database? At the moment, the licensing conditions of several databases prevent integrated databases to be made available as a whole. As a compromise, the Columba database provides data for single entries in the XML format.

At the moment, Columba offers a superior range of query options and integrated data sources compared to other integrated structure databases. However, competitors like the reengineered PDB site are catching up, meaning that the contest for the best web interface is likely to continue. For the reasons given above, the data warehouse itself will be highly valuable for the Columba developers and their fellow scientists apart from the website.

12 Protein structures are packed more loosely where conformational flexibility is required

The analysis of protein packing is one of the fields, where the Columba database can be applied. It was used to establish clearly defined and representative datasets of protein structures effectively. Using an improved Voronoi procedure, packing was studied in membrane coils and membrane gates; coenzymes and small organic ligands; their binding sites; and in reference data derived from SCOP.

Packing in different depths from the protein surface. In the reference data, packing was tracked from the protein surface to the most occluded atoms in order to resolve contradictions in the literature about the localization of cavities in proteins. Using the ProtOr radii set and ProtOr atom types, the distance to the protein surface was used to define regions in the protein interior and to analyse the occurrence of cavities. Unlike previous approaches, this method takes every protrusion and groove in the structure into account, but ignores internal cavities, which is a necessity for measuring their frequency.

The initial question, how different regions are packed, has been answered. It has been found that packing in the protein interior is tight but generally inhomogenous. The well-packed SHALLOW subsets are located around the DEEP subset, in which packing is slightly less tight. By removing two layers of surface atoms, the BOTTOM subset is gained, which contains mostly the tightly packed atoms from the other two subsets, and consequently is more spread out.

In proteins, atom-sized cavities occur in all distances from the surface. There are about 4.4 cavities per 100 amino acids in protein structures; they occur in all regions, most frequently in a depth of 2.74-3.52Å underneath the superficial atoms, while the cavity neighboring atoms are observed most frequently in the protein core [Rother et al., 2003]. This discrepancy is due to cavity neighboring atoms being overrepresented far from the surface, because relatively few protein atoms exist there. Around cavities, packing is less efficient. The composition of cavity neighboring atoms is more nonpolar than the corresponding subset. Cavities are slightly less polar towards the protein core. The reference packing densities presented in table 10.1 for several protein regions, provide a basis for assessing the packing in other structures and for identifying tightly packed regions and cavities.

Packing in membrane helices. Packing in 20 transmembrane domains was also analyzed. It was found that a group of 10 channels and transporters is packed more loosely than helix-helix interfaces from other membrane domains and non-membrane proteins [Hildebrand et al., 2005]. Also, the channels/transporters exhibit internal cavities in interesting positions: Polar cavities preferably situate themselves at locations, at which a channel accomodates the passage of polar molecules through the protein, while nonpolar cavities are often found in hinge regions, increasing their flexibility.

Packing of coenzymes, small organic molecules, and binding sites in proteins. It was found that small organic molecules and coenzymes are packed as tightly as comparable protein atoms, or even better. Only some aromatic groups in the coenzymes have a lower packing density. A higher number of atom-sized cavities (6.9 per kDa ligand atoms and 9.2 per kDa ligand neighbors) than in the reference dataset was discovered. With most of them filled with water *in vivo*, they are supposed to mediate the protein-ligand

interaction. A reasonable interpretation is that the protein ligands studied here must be very close to the protein atoms of the binding site, because the neighboring cavities do not lower the average packing density. This interpretation corresponds well with the packing of protein atoms in the binding pocket, which is tighter than in the reference data.

When considered jointly, these results indicate that internal cavities occur frequently there, where local flexibility is required for proper functionality or the structure needs to accomodate the binding of a particular molecule.

Part V

Appendix

Bibliography

F. H. Allen. The Cambridge Structural Database: a quarter of a million crystal structures and rising. *Acta Crystallogr B*, 58:380–308, 2002.

J. An, T. Nakama, Y. Kubota, and A. Sarai. 3DinSight: an integrated relational database and search tool for the structure, function and properties of biomolecules. *Bioinformatics*, 14(2):188 – 195, 1998. URL `http://www.rtc.riken.go.jp/3DinSight.html`.

A. Andreeva, D. Howorth, S. E. Brenner, T. J. P. Hubbard, C. Chothia, and A. G. Murzin. SCOP database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Res*, 32:D226 – D229, 2004. URL `http://scop.mrc-lmb.cam.ac.uk/scop`. Database issue.

M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. Matese, J. Richardson, M. Ringwald, G. Rubin, and G. Sherlock. Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1): 25–29, 2000. doi: 10.1038/75556. URL `http://dx.doi.org/10.1038/75556`.

Z. Bagci, A. Kloczkowski, R. L. Jernigan, and I. Bahar. The origin and extent of coarse-grained regularities in protein internal packing. *Proteins*, 53(1):56–67, 2003. doi: 10.1002/prot.10435. URL `http://dx.doi.org/10.1002/prot.10435`.

A. Bairoch. The ENZYME database in 2000. *Nucleic Acids Res*, 28:304–305, 2000.

A. Bairoch, R. Apweiler, C. H. Wu, W. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, L. R., M. Magrane, M. Martin, D. Natale, C. ODonovan, N. Redaschi, and L. Yeh. The Universal Protein Resource (UniProt). *Nucleic Acids Res*, 33:D154–159, 2005. doi: 10.1093/nar/gki070. URL `http://dx.doi.org/10.1093/nar/gki070`. Database issue.

P. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens. Tambis: Transparent access to multiple bioinformatics information sources. *6th Int. Conf. on Intelligent Systems for Molecular Biology, Montreal, Canada: AAAI Press, Menlo Park*, pages 25–34, 1998.

N. Ban, P. Nissen, J. Hansen, P. Moore, and T. Steitz. The complete atomic structure of the large ribosomal subunit at 2.4 å resolution. *Science*, 289(5481):905–920, 2000.

H. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide Protein Data Bank. *Nat Struct Biol*, 10(12):980, 2003. doi: 10.1038/nsb1203-980. URL `http://dx.doi.org/10.1038/nsb1203-980`.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res*, 28(1): 235–242, 2000. URL `http://www.rcsb.org/pdb/`.

H. M. Berman, T. Battistuz, T. N. Bhat, W. F. Bluhm, P. E. Bourne, K. Burkhardt, Z. Feng, G. L. Gilliland, L. Iype, S. Jain, P. Fagan, J. Marvin, D. Padilla, V. Ravichandran, B. Schneider, N. Thanki, H. Weissig, J. D. Westbrook, and C. Zardecki. The Protein Data Bank. *Acta Crystallogr D Biol Crystallogr*, 58:899–907, 2002a.

H. M. Berman, J. Westbrook, Z. Feng, L. Iype, B. Schneider, and C. Zardecki. The Nucleic Acid Database. *Acta Crystallogr D Biol Crystallogr*, 58(6/1):889–898, 2002b.

T. Bhat, P. Bourne, Z. Feng, G. Gilliland, S. Jain, V. Ravichandran, B. Schneider, K. Schneider, N. Thanki, H. Weissig, J. Westbrook, and H. M. Berman. The PDB data uniformity project. *Nucleic Acids Res*, 29(1):214 – 218, 2001. URL `http://www.rcsb.org/pdb`.

A. Birkland and G. Yona. Biozon: a hub of heterogeneous biological data. *Nucleic Acids Res*, 34:D235 – D242, 2006. Database issue.

B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res*, 31(1):365–370, 2003. URL `http://www.expasy.org/sprot/`.

H. Boutselakis, D. Dimitropoulos, J. Fillon, A. Golovin, K. Henrick, A. Hussain, J. Ionides, M. John, P. A. Keller, E. Krissinel, P. McNeil, A. Naim, R. Newman, T. Oldfield, J. Pineda, A. Rachedi, J. Copeland, A. Sitnov, S. Sobhany, A. Suarez-Uruena, J. Swaminathan, M. Tagari, J. Tate, S. Tromm, S. Velankar, and W. Vranken. E-MSD: the European Bioinformatics Institute Macromolecular Structure Database. *Nucleic Acids Res*, 31:458–462, 2003.

B. Brooks, R. Bruccoleri, B. Olafson, D. States, S. Swaminathan, , and M. Karplus. CHARMM: A program for macromolecular energy, minmimization, and dynamics calculations. *J Comp Chem*, 4:187 – 217, 1983.

F. Brooks. *The mythical man-month*. Addison-Wesley, 1975.

Y. V. Bukhman and J. Skolnick. BioMolQuest: integrated database-based retrieval of protein structural and functional information. *Bioinformatics*, 17(5):468 – 478, 2001.

E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez, and R. Apweiler. The Gene Ontology Annotation (GOA) Database: sharing knowledge in UniProt with Gene Ontology. *Nucleic Acids Res*, 32:D262 – D266, 2004. doi: http://www.geneontology.org.

O. Carugo and S. Pongor. The evolution of structural databases. *Trends Biotechnol*, 20: 498–501, 2002.

D. U. Chaudhuri S. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26:65–74, 1997.

C. Chothia. Structural invariants in protein folding. *Nature*, 254:304–308, 1975.

C. Chothia and A. Lesk. The relation between the divergence of sequence and structure in proteins. *EMBO J*, 5(4):823–826, 1986.

C. Chothia, M. Levitt, and D. Richardson. Helix to helix packing in proteins. *J Mol Biol*, 145(1):215–50, Jan 1981.

M. Connolly. Solvent-accessible surfaces of proteins and nucleic acids. *Science*, 19 (221(4612)):709–713, 1983.

G. O. Consortium. Creating the Gene Ontology resource: design and implementation. *Genome Res*, 11(8):1425–1433, 2001.

G. O. Consortium. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res*, 32:D258 – D261, 2004. URL `http://www.geneontology.org`. Database issue.

B. Dahiyat and S. Mayo. Probing the role of packing specificity in protein design. *Proc Natl Acad Sci USA*, 94(19):10172–10177, 1997.

W. F. DeGrado, H. Gratkowski, and J. D. Lear. How do helix-helix interactions help determine the folds of membrane proteins? Perspectives from the study of homo-oligomeric helical bundles. *Protein Sci*, 12(4):647–665, 2003.

J. Deisenhofer, O. Epp, K. Miki, R. Huber, and H. Michel. X-ray structure analysis of a membrane protein complex. Electron density map at 3 å resolution and a model of the chromophores of the photosynthetic reaction center from Rhodopseudomonas viridis. *J Mol Biol*, 180(2):385–398, 1984.

K. A. Denessiouk and M. S. Johnson. "Acceptor-donor-acceptor" motifs recognize the Watson-Crick, Hoogsteen and Sugar "donor-acceptor-donor" edges of adenine and adenosine-containing ligands. *J Mol Biol*, 333(5):1025–1043, 2003.

A. I. Denesyuk, K. A. Denessiouk, T. Korpela, and M. S. Johnson. Phosphate group binding "cup" of PLP-dependent and non-PLP-dependent enzymes: leitmotif and variations. *Biochim Biophys Acta*, 1647(1-2):234–238, 2003.

G. Dieterich, M. Plail, W.-D. Schubert, and J. Reichel. Raptor3D: a tool for automatic mapping of up-to-date functional annotations to three-dimensional protein structures. *J Appl Cryst*, 38:856–857, 2005.

S. Dietmann, J. Park, C. Notredame, A. Heger, M. Lappe, and L. Holm. A fully automatic evolutionary classification of protein folds: Dali Domain Dictionary version 3. *Nucleic Acids Res*, 29(1):55–57, 2001.

H. Do and E. Rahm. Coma - a system for flexible combination of schema matching approaches. Conference on Very Large Data Bases(VLDB), 2002.

C. Dodge, R. Schneider, and C. Sander. The HSSP database of protein structure-sequence alignments and family profiles. *Nucleic Acids Res*, 26(1):313–315, 1998.

R. Dowell, R. Jokerst, A. Day, S. Eddy, and L. Stein. The distributed annotation system. *BMC Bioinformatics*, 2:7, 2001.

N. Echols, D. Milburn, and M. Gerstein. MolMovDB: analysis and visualization of conformational change and structural flexibility. *Nucleic Acids Res*, 31(1):478–482, 2003.

M. Eilers, A. B. Patel, W. Liu, and S. O. Smith. Comparison of helix interactions in membrane and soluble alpha-bundle proteins. *Biophys J*, 82(5):2720–36, May 2002.

A. E. Eriksson, W. A. Baase, X. J. Zhang, D. W. Heinz, M. Blaber, E. P. Baldwin, and B. W. Matthews. Response of a protein structure to cavity-creating mutations and its relation to the hydrophobic effect. *Science*, 255:178–183, 1992.

Z. Feng, L. Chen, H. Maddula, O. Akcan, R. Oughtred, H. M. Berman, and J. Westbrook. Ligand Depot: a data warehouse for ligands bound to macromolecules. *Bioinformatics*, 20(13):2153–2155, 2004. doi: 10.1093/bioinformatics/bth214. URL `http://dx.doi.org/10.1093/bioinformatics/bth214`.

J. Finney. Random packings and the structure of simple liquids. i. the geometry of random close packing. *Proc Roy Soc ser A*, 319:479 – 493, 1970.

S. J. Fleishman and N. Ben-Tal. A novel scoring function for predicting the conformations of tightly packed pairs of transmembrane alpha-helices. *J Mol Biol*, 321(2):363–378, 2002.

P. J. Fleming and F. M. Richards. Protein packing: dependence on protein size, secondary structure and amino acid composition. *J Mol Biol*, 299:487–498, 2000.

G. B. Fogel, D. G. Weekes, G. Varga, E. R. Dow, A. M. Craven, H. B. Harlow, E. W. Su, J. E. Onyia, and C. Su. A statistical analysis of the TRANSFAC database. *Biosystems*, 81(2):137–154, 2005. doi: 10.1016/j.biosystems.2005.03.003. URL `http://dx.doi.org/10.1016/j.biosystems.2005.03.003`.

B. J. Gellatly and J. L. Finney. Calculation of protein volumes: An alternative to the Voronoi procedure. *J Mol Biol*, 161:305–322, 1982.

M. Gerstein and C. Chothia. Packing at the protein-water interface. *Proc Natl Acad Sci USA*, 93:10167–10172, 1996.

M. Gerstein, J. Tsai, and M. Levitt. The volume of atoms on the protein surface: calculated from simulation, using Voronoi polyhedra. *J Mol Biol*, 249:955–966, 1995.

C. Gille, S. Lorenzen, E. Michalsky, and C. Frömmel. KISS for STRAP: user extensions for a protein alignment editor. *Bioinformatics*, 19(18):2489–2491, 2003.

A. Goede, R. Preissner, and C. Frömmel. Voronoi cell: New method for allocation of space among atoms: Elimination of avoidable errors in calculation of atomic volume and density. *J Comput Chem*, 18:1113–1123, 1997.

A. Goede, M. Dunkel, N. Mester, C. Frömmel, and R. Preissner. SuperDrug: a conformational drug database. *Bioinformatics*, 21(9):1751–1753, 2005. doi: 10.1093/bioinformatics/bti295. URL `http://dx.doi.org/10.1093/bioinformatics/bti295`.

A. Golovin, T. J. Oldfield, J. G. Tate, S. Velankar, G. J. Barton, H. Boutselakis, D. Dimitropoulos, J. Fillon, A. Hussain, J. M. C. Ionides, M. John, P. A. Keller, E. Krissinel, P. McNeil, A. Naim, R. Newman, A. Pajon, J. Pineda, A. Rachedi, J. Copeland, A. Sitnov, S. Sobhany, A. Suarez-Uruena, G. J. Swaminathan, M. Tagari, S. Tromm, W. Vranken, and . Henrick, K. E-MSD: an integrated data resource for bioinformatics. *Nucleic Acids Res*, 32:D211 – D216, 2004. URL `http://www.ebi.ac.uk/msd/`.

D. S. Greer, J. D. Westbrook, and P. E. Bourne. An ontology driven architecture for derived representations of macromolecular structure. *Bioinformatics*, 18(9):1280 – 1281, 2002. URL `http://openmms.sdsc.edu`.

S. Günther, K. Rother, and C. Frömmel. Molecular flexibility in protein-dna interactions. *Biosystems*, 2006. Epub ahead of print.

A. Gutteridge and J. Thornton. Conformational changes observed in enzyme crystal structures upon substrate binding. *J Mol Biol*, 346(1):21–28, 2005. doi: 10.1016/j.jmb.2004.11.013. URL `http://dx.doi.org/10.1016/j.jmb.2004.11.013`.

C. Hadley and D. T. Jones. A systematic comparison of protein structure classifications: Scop, cath and fssp. *Structure Fold Des*, 7:1099–1112, 1999.

I. Halperin, H. Wolfson, and R. Nussinov. Protein-protein interactions; coupling of structurally conserved residues and of hot spots across interfaces. Implications for docking. *Structure (Camb)*, 12(6):1027–1038, 2004. doi: 10.1016/j.str.2004.04.009. URL `http://dx.doi.org/10.1016/j.str.2004.04.009`.

T. Hamelryck and B. Manderick. PDB file parser and structure class implemented in Python. *Bioinformatics*, 19(17):2308–2310, 2003.

Y. Harpaz, M. Gerstein, and C. Chothia. Volume changes on protein folding. *Structure*, 2:641–649, 1994.

M. Hendlich, A. Bergner, J. Günther, and G. Klebe. Relibase: design and development of a database for comprehensive analysis of protein-ligand interactions. *J Mol Biol*, 326(2):607–620, 2003.

P. W. Hildebrand, R. Preissner, and C. Frömmel. Structural features of transmembrane helices. *FEBS Lett*, 559(1-3):145–151, 2004. doi: 10.1016/S0014-5793(04)00061-4. URL `http://dx.doi.org/10.1016/S0014-5793(04)00061-4`.

P. W. Hildebrand, K. Rother, A. Goede, R. Preissner, and C. Frömmel. Molecular packing and packing defects in helical membrane proteins. *Biophys J*, 88(3):1970–1977, 2005. doi: 10.1529/biophysj.104.049585. URL `http://dx.doi.org/10.1529/biophysj.104.049585`.

U. Hobohm, M. Scharf, R. Schneider, and C. Sander. Selection of representative protein data sets. *Protein Sci*, 1(3):409–417, 1992.

R. Hooft, G. Vriend, C. Sander, and E. Abola. Errors in protein structures. *Nature*, 381 (6580):272, 1996.

H. Huang, W. C. Barker, Y. Chen, and H. Wu, Cathy. iProClass: an integrated database of protein family, function and structure information. *Nucleic Acids Res*, 31(1):390 – 392, 2003. URL `http://pir.georgetown.edu/iproclass`.

S. J. Hubbard, K. H. Gross, and P. Argos. Intramolecular cavities in globular proteins. *Protein Eng*, 7:613–626, 1994.

K. Ishikawa, H. Nakamura, K. Morikawa, and S. Kanaya. Stabilization of Escherichia coli ribonuclease HI by cavity-filling mutations within a hydrophobic core. *Biochemistry*, 32:6171–6178, 1993.

Y. Jiang, A. Lee, J. Chen, M. Cadene, B. T. Chait, and R. MacKinnon. The open pore conformation of potassium channels. *Nature*, 417(6888):523–526, 2002. doi: 10.1038/417523a. URL `http://dx.doi.org/10.1038/417523a`.

W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.

M. Kanehisa, S. Goto, S. Kavashima, Y. Okuno, and M. Hattori. The KEGG resource for deciphering the genome. *Bioinformatics*, 32:D277 – D280, 2004. URL `http://www.genome.ad.jp/kegg/`. Database issue.

A. Karshikoff and R. Ladenstein. Proteins from thermophilic and mesophilic organisms essentially do not differ in packing. *Protein Eng*, 11(10):867–872, 1998.

A. Kasprzyk, D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, M. Hammond, P. Rocca-Serra, T. Cox, and E. Birnex. EnsMart: A Generic System for Fast and Flexible Access to Biological Data. *Genom Research*, 14:160 – 169, 2004. URL `http://www.ensembl.org/EnsMart`.

G. J. Kleywegt and T. A. Jones. Homo crystallographicus–quo vadis? *Structure (Camb)*, 10(4):465–472, 2002.

J. P. Kocher, M. Prevost, S. J. Wodak, and B. Lee. Properties of the protein matrix revealed by the free energy of cavity formation. *Structure*, 4:1517–1529, 1996.

H. Kono, M. Nishiyama, M. Tanokura, and J. Doi. Designing the hydrophobic core of Thermus flavus malate dehydrogenase based on side-chain packing. *Protein Eng*, 11 (1):47–52, 1998.

A. Krause, J. Stoye, and M. Vingron. The SYSTERS protein sequence cluster set. *Nucleic Acids Res*, 28(1):270–272, 2000. URL `http://www.dkfz-heidelberg.de/tbi/services/cluster/systersform`.

L. A. Kuhn, M. A. Siani, M. E. Pique, C. L. Fisher, E. D. Getzoff, and J. A. Tainer. The interdependence of protein surface topography and bound water molecules revealed by surface accessibility and fractal density measures. *J Mol Biol*, 228:13–12, 1992.

L. V. S. Lakshmanan, F. Sadri, and I. N. Subramanian. On the logical foundation of schema integration and evolution in heterogeneous database systems. In *2nd Int. Conf. on Deductive and Object-Oriented Databases*, pages 81–100, Phoenix, Arizona, 1993.

D. Langosch and J. Heringa. Interaction of transmembrane helices by a knobs-into-holes packing characteristic of soluble coiled coils. *Proteins*, 31(2):150–159, 1998.

R. A. Laskowski, V. V. Chistyakov, and J. M. Thornton. PDBsum more: new summaries and analyses of the known 3D structures of proteins and nucleic acids. *Nucleic Acids Res*, 33(Database issue):D266–268, 2005. doi: 10.1093/nar/gki001.

U. Leser and F. Naumann. (almost) hands-off information integration for the life sciences. In *Conference on Innovative Database Research (CIDR)*, 2005.

W. Li, L. Jaroszewski, and A. Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, 17(3):282–283, 2001.

J. Liang and K. Dill. Are proteins well-packed? *Biophys J*, 81(2):751–766, 2001.

J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical shape computation of macromolecules: II. inaccessible cavities in proteins. *Proteins*, 33:18–29, 1998a.

J. Liang, H. Edelsbrunner, and C. Woodward. Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. *Protein Sci*, 7:1884–1897, 1998b.

H. Liao, W. Yeh, D. Chiang, R. Jernigan, and B. Lustig. Protein sequence entropy is closely related to packing density and hydrophobicity. *Protein Eng Des Sel*, 18(2): 59–64, 2005. doi: 10.1093/protein/gzi009.

W. Liu, M. Eilers, A. B. Patel, and S. O. Smith. Helix packing moments reveal diversity and conservation in membrane protein structure. *J Mol Biol*, 337(3):713–729, 2004a. doi: 10.1016/j.jmb.2004.02.001. URL `http://dx.doi.org/10.1016/j.jmb.2004.02.001`.

X. Liu, K. Fan, and W. Wang. The number of protein folds and their distribution over families in nature. *Proteins*, 54(3):491–499, 2004b. doi: 10.1002/prot.10514. URL `http://dx.doi.org/10.1002/prot.10514`.

K. P. Locher, R. B. Bass, and D. C. Rees. Structural biology. Breaching the barrier. *Science*, 301(5633):603–604, 2003. doi: 10.1126/science.1088621. URL `http://dx.doi.org/10.1126/science.1088621`.

V. V. Loladze, D. N. Ermolenko, and G. I. Makhatadze. Thermodynamic consequences of burial of polar and non-polar amino acid residues in the protein interior. *J Mol Biol*, 320(2):343–357, 2002. doi: 10.1016/S0022-2836(02)00465-5. URL `http://dx.doi.org/10.1016/S0022-2836(02)00465-5`.

J. Löwe, D. Stock, B. Jap, P. Zwickl, W. Baumeister, and R. Huber. Crystal structure of the 20S proteasome from the archaeon T. acidophilum at 3.4 å resolution. *Science*, 268(5210):533–539, 1995.

C. Machicado, M. Bueno, and J. Sancho. Predicting the structure of protein cavities created by mutation. *Protein Eng*, 15(8):669–675, 2002.

K. MacKenzie and D. Engelman. Structure-based prediction of the stability of transmembrane helix-helix interactions: the sequence dependence of glycophorin A dimerization. *Proc Natl Acad Sci USA*, 95(7):3583–3590, 1998.

H. Mangalam. The Bio* toolkits–a brief overview. *Brief Bioinform*, 3(3):296–302, 2002.

A. C. R. Martin. PDBSprotEC: a Web-accessible database linking PDB chains to EC numbers via SwissProt. *Bioinformatics*, 20(6):986–988, Apr 2004. doi: 10.1093/bioinformatics/bth048. URL `http://dx.doi.org/10.1093/bioinformatics/bth048`.

M. Matsumura, W. J. Becktel, and B. W. Matthews. Hydrophobic stabilization in T4 lysozyme determined directly by multiple substitutions of Ile 3. *Nature*, 334:406–410, 1988.

P. May, S. Barthel, and I. Koch. PTGL–a web-based database application for protein topologies. *Bioinformatics*, 20(17):3277–3279, 2004. doi: 10.1093/bioinformatics/bth367. URL `http://dx.doi.org/10.1093/bioinformatics/bth367`.

G. Michal. Biochemical Pathways. Boehringer Mannheim GmbH, 1993. URL `http://www.expasy.org/cgi-bin/search-biochem-index`.

E. Michalsky, M. Dunkel, A. Goede, and R. Preissner. SuperLigands - a database of ligand structures derived from the Protein Data Bank. *BMC Bioinformatics*, 6 (1):122, 2005. doi: 10.1186/1471-2105-6-122. URL `http://dx.doi.org/10.1186/1471-2105-6-122`.

N. J. Mulder, R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bradley, P. Bork, P. Bucher, L. Cerutti, R. Copley, E. Courcelle, U. Das, R. Durbin, W. Fleischmann, J. Gough, D. Haft, N. Harte, N. Hulo, D. Kahn, A. Kanapin, M. Krestyaninova, D. Lonsdale, R. Lopez, I. Letunic, M. Madera, J. Maslen, J. McDowall, A. Mitchell, A. N. Nikolskaya, S. Orchard, M. Pagni, C. P. Ponting, E. Quevillon, J. Selengut, C. J. A. Sigrist, V. Silventoinen, D. J. Studholme, R. Vaughan, and C. H. Wu. InterPro, progress and status in 2005. *Nucleic Acids Res*, 33(Database issue):D201–205, 2005. doi: 10.1093/nar/gki106. URL `http://dx.doi.org/10.1093/nar/gki106`.

A. Murzin, S. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247 (4):536–540, 1995. doi: 10.1006/jmbi.1995.0159. URL `http://scop.mrc-lmb.cam.ac.uk/scop`.

K. Nadassy, I. Tomas-Oliveira, I. Alberts, J. Janin, and S. J. Wodak. Standard atomic volumes in double-stranded DNA and packing in protein–DNA interfaces. *Nucleic Acids Res*, 29:3362–3376, 2001.

K. Ogata, C. Kanei-Ishii, M. Sasaki, H. Hatanaka, A. Nagadoi, M. Enari, H. Nakamura, Y. Nishimura, S. Ishii, and A. Sarai. The cavity in the hydrophobic core of Myb DNA-binding domain is reserved for DNA recognition and trans-activation. *Nat Struct Biol*, 3:178–187, 1996.

C. Orengo, A. Michie, S. Jones, D. Jones, M. Swindells, and J. Thornton. CATH–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108, 1997.

E. Paci and M. Marchi. Intrinsic compressibility and volume compression in solvated proteins by molecular dynamics simulation at high pressure. *Proc Natl Acad Sci USA*, 93(21):11609–11614, 1996.

N. Paton, S. Khan, A. Hayes, F. Moussouni, A. Brass, K. Eilbeck, C. Goble, S. Hubbard, and S. Oliver. Conceptual modelling of genomic information. *Bioinformatics*, 16(6): 548–557, 2000.

N. Pattabiraman, K. B. Ward, and P. J. Fleming. Occluded molecular surface: analysis of protein packing. *J Mol Recognit*, 8:334–344, 1995.

F. Pearl, A. Todd, I. Sillitoe, M. Dibley, O. Redfern, T. Lewis, C. Bennett, R. Marsden, A. Grant, D. Lee, A. Akpor, M. Maibaum, A. Harrison, T. Dallman, G. Reeves, I. Diboun, S. Addou, S. Lise, C. Johnston, A. Sillero, J. Thornton, and C. Orengo. The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucleic Acids Res*, 33 (Database issue):D247–251, 2005. doi: 10.1093/nar/gki024.

E. Perozo, D. M. Cortes, P. Sompornpisut, A. Kloda, and B. Martinac. Open channel structure of MscL and the gating mechanism of mechanosensitive channels. *Nature*, 418(6901):942–8, Aug 2002. doi: 10.1038/nature00992.

J. Pontius, J. Richelle, and S. J. Wodak. Deviations from standard atomic volumes as a quality measure for protein crystal structures. *J Mol Biol*, 264:121–136, 1996.

J. Popot and D. Engelman. Helical membrane protein folding, stability, and evolution. *Annu Rev Biochem*, 69:881–922, 2000. doi: 10.1146/annurev.biochem.69.1.881.

C. T. Porter, G. J. Bartlett, and J. M. Thornton. The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res*, 32(Database issue):D129–133, 2004. doi: 10.1093/nar/gkh028.

R. Preissner, A. Goede, and C. Frömmel. Dictionary of interfaces in proteins (dip). data bank of complementary molecular surface patches. *J Mol Biol*, 280:535–550, 1998.

E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, 10(4):334–350, 2001.

V. Rantanen, K. Denessiouk, M. Gyllenberg, T. Koski, and M. Johnson. A fragment library based on Gaussian mixtures predicting favorable molecular interactions. *J Mol Biol*, 313(1):197–214, 2001. doi: 10.1006/jmbi.2001.5023.

V.-V. Rantanen, M. Gyllenberg, T. Koski, and M. S. Johnson. A Bayesian molecular interaction library. *J Comput Aided Mol Des*, 17(7):435–461, 2003.

M. Rebhan, V. Chalifa-Caspi, J. Prilusky, and D. Lancet. GeneCards: a novel functional genomics compendium with automated data mining and query reformulation support. *Bioinformatics*, 14:656–664, 1998.

J. Reichelt, G. Dieterich, M. Kvesic, D. Schomburg, and D. W. Heinz. BRAGI: linking and visualization of database information in a 3D viewer and modeling tool. *Bioinformatics*, 21(7):1291–1293, 2005. doi: 10.1093/bioinformatics/bti138.

J. Reichert and J. Sühnel. The imb jena image library of biological macromolecules: 2002 update. *Nucleic Acids Res*, 30:253–254, 2002.

F. M. Richards. The interpretation of protein structures: total volume, group volume distributions and packing density. *J Mol Biol*, 82:1–4, 1974.

T. J. Richmond. Solvent accessible surface area and excluded volume in proteins. analytical equations for overlapping spheres and implications for the hydrophobic effect. *J Mol Biol*, 178:63–89, 1984.

O. Ritter, P. Kocab, M. Senger, D. Wolf, and S. Suhai. Prototype implementation of the integrated genomic database. *Comput Biomed Res*, 27(2):97–115, 1994.

K. Rother. Packungsdichte und Packungsdefekte in Proteinstrukturen. Master's thesis, FU Berlin, 2002.

K. Rother, R. Preissner, A. Goede, and C. Frömmel. Inhomogeneous molecular density: reference packing densities and distribution of cavities within proteins. *Bioinformatics*, 19(16):2112–2121, 2003.

K. Rother, H. Müller, S. Trissl, I. Koch, T. Steinke, R. Preissner, C. Frömmel, and U. Leser. Columba: multidimensional data integration of protein annotations. In *DILS*, volume 2994 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 2004. ISBN 3-540-21300-7.

K. Rother, E. Michalsky, and U. Leser. How well are protein structures annotated in secondary databases? *Proteins*, 60(4):571–576, 2005. doi: 10.1002/prot.20520.

K. Rother, M. Dunkel, E. Michalsky, S. Trissl, A. Goede, U. Leser, and R. Preissner. A structural keystone for drug design. *Journal of Integrative Bioinformatics*, 0019, 2006. Online Journal.

W. S. Sandberg and T. C. Terwilliger. Influence of interior packing and hydrophobicity on the stability of a protein. *Science*, 245:54–57, 1989.

I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, and D. Schomburg. BRENDA, the enzyme database: updates and major new developments. *Nucleic Acids Res*, 32:D431–433, 2004. doi: 10.1093/nar/gkh081. Database issue.

A. Senes, I. Ubarretxena-Belandia, and D. Engelman. The Calpha —H...O hydrogen bond: a determinant of stability and specificity in transmembrane helix interactions. *Proc Natl Acad Sci USA*, 98(16):9056–9061, 2001. doi: 10.1073/pnas.161280798.

A. J. Shepherd, N. J. Martin, R. G. Johnson, P. Kellam, and C. A. Orengo. PFDB: a generic protein family database integrating the CATH domain structure database with sequence based protein family resources. *Bioinformatics*, 18(12):1666–1672, 2002.

I. Shindyalov and P. Bourne. A database and tools for 3-D protein structure comparison and alignment using the Combinatorial Extension (CE) algorithm. *Nucleic Acids Res*, 29(1):228–229, 2001.

P. H. A. Sneath and R. R. Sokal. Numerical taxonomy: the principles and practices of numerical classification. *W. H. Freeman, San Francisco*, 1:230–234, 1973.

J. E. Stajich, D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, G. Fuellen, J. G. R. Gilbert, I. Korf, H. Lapp, H. Lehväslaiho, C. Matsalla, C. J. Mungall, B. I. Osborne, M. R. Pocock, P. Schattner, M. Senger, L. D. Stein, E. Stupka, M. D. Wilkinson, and E. Birney. The Bioperl toolkit: Perl modules for the life sciences. *Genome Res*, 12(10):1611–1618, 2002. doi: 10.1101/gr.361602.

L. Stein. Creating a bioinformatics nation. *Nature*, 417(6885):119–120, 2002. doi: 10.1038/417119a.

L. D. Stein. Integrating biological databases. *Nat Rev Genet*, 4:337–345, 2003.

P. F. W. Stouten, C. Frömmel, H. Nakamura, and C. Sander. An effective solvation term based on atomic occupancies for use in protein. *Mol Simul*, 10:97–120, 1993.

A. C. Stuart, V. A. Ilyin, and A. Sali. LigBase: a database of families of aligned ligand binding sites in known protein sequences and structures. *Bioinformatics*, 18(1):200–201, 2002.

K. J. Swartz. Opening the gate in potassium channels. *Nat Struct Mol Biol*, 11(6):499–501, 2004. doi: 10.1038/nsmb0604-499.

A. Szilagyi and P. Zavodszky. Structural differences between mesophilic, moderately thermophilic and extremely thermophilic protein subunits: results of a comprehensive survey. *Structure Fold Des*, 8:493–504, 2000.

S. Trissl and U. Leser. Querying ontologies in relational database systems. In *2nd Conference on Data Integration in the Life Sciences (DILS05), San Diego, USA*, volume 3615, pages 63–79, 2005.

S. Trissl, K. Rother, H. Müller, T. Steinke, I. Koch, R. Preissner, C. Frömmel, and U. Leser. Columba: an integrated database of proteins, structures, and annotations. *BMC Bioinformatics*, 6(1):81, 2005. doi: 10.1186/1471-2105-6-81.

J. Tsai and M. Gerstein. Calculations of protein volumes: sensitivity analysis and parameter database. *Bioinformatics*, 18(7):985–995, 2002.

J. Tsai, R. Taylor, C. Chothia, and M. Gerstein. The packing density in proteins: standard radii and volumes. *J Mol Biol*, 290:253–266, 1999.

J. Tsai, N. Voss, and M. Gerstein. Determining the minimum number of types necessary to represent the sizes of protein atoms. *Bioinformatics*, 17:949–956, 2001.

S. Velankar, P. McNeil, V. Mittard-Runte, A. Suarez, D. Barrell, R. Apweiler, and K. Henrick. E-MSD: an integrated data resource for bioinformatics. *Nucleic Acids Res*, 33 (Database issue):D262–265, 2005. doi: 10.1093/nar/gki058.

G. F. Voronoi. Nouvelles applications des parametres continus e la theorie de formes quadratiques. *J Reine Angew Math*, 134:198–287, 1908.

N. Voss and M. Gerstein. Calculation of standard atomic volumes for RNA and comparison with proteins: RNA is packed more tightly. *J Mol Biol*, 346(2):477–492, 2005. doi: 10.1016/j.jmb.2004.11.072.

G. Wang and R. L. Dunbrack Jr. PISCES: a protein sequence culling server. *Bioinformatics*, 19(12):1589–1591, 2003.

J. Westbrook, N. Ito, H. Nakamura, K. Henrick, and H. M. Berman. PDBML: the representation of archival macromolecular structure data in XML. *Bioinformatics*, 21 (7):988–992, 2005. doi: 10.1093/bioinformatics/bti082.

D. Wheeler, C. Chappey, A. Lash, D. Leipe, T. Madden, G. Schuler, T. Tatusova, and B. Rapp. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res*, 28(1):10–14, 2000.

S. White and W. Wimley. Membrane protein folding and stability: physical principles. *Annu Rev Biophys Biomol Struct*, 28:319–365, 1999. doi: 10.1146/annurev.biophys.28.1.319.

E. M. Zdobnov, R. Lopez, R. Apweiler, and T. Etzold. The EBI SRS server-new features. *Bioinformatics*, 18(8):1149–1150, 2002a.

E. M. Zdobnov, R. Lopez, R. Apweiler, and T. Etzold. The EBI SRS server–recent developments. *Bioinformatics*, 18:368–373, 2002b.

J. Zhang, Y. Chen, R. Chen, and J. Liang. Importance of chirality and reduced flexibility of protein side chains: a study with square and tetrahedral lattice models. *J Chem Phys*, 121(1):592–603, 2004. doi: 10.1063/1.1756573.

A Description of data sources integrated in the Columba database

1 Data sources providing protein structures

A.1.1 PDB

Name	PDB - Protein Structure Database Berman et al. [2000], recent updates in Berman et al. [2002a, 2003]
URL	http://www.pdb.org
Description	The PDB contains protein structures, the basis for Columba. In Columba, only the description from the PDB header, but not the structure itself is included. Each entry is qualified further by the *compounds* defined in the header, by polymer *chains* including the amino acid sequence parsed from the atom records, and non-polymer molecules, the *hetero* groups. A PDB entry does not necessarily have *chain* and *hetero* records, but they typically have both.
Data source ID	pdb
Requires	-
DB schema	columba
DB tables	pdb_entry, compound, chain, hetero
Cross-references	Are not used at the moment. The reason for disregarding links from PDB entries to second-party databases is the very nature of PDB, which essentially is an uncurated archive of structures. Responsibility of the content of an entry is by the authors, not by the PDB. Therefore, links are often not updated after the initial submission. Note that PDB is currently changing this policy at least in part, and there are attempts to curate PDB entries in certain aspects Velankar et al. [2005], Andreeva et al. [2004].
Primary data	ASCII files in PDB format (about 10 GB)
Updates	Almost daily, weekly updates are announced by the PDB.
Parser	Selfmade Python parser that uses some AWK scripts by Andreas Hoppe. Formerly, BioPython::Bio.PDB.PDBParser was used for parsing chains and hetero groups, but found to fail for some structures.
Available on website	Yes
Module author	Kristian Rother

A.2.2 Swiss-Prot / Gene Ontology

Name	Swiss-Prot - Protein knowledgebase Boeckmann et al. [2003], GO - Gene Ontology Ashburner et al. [2000], updates in Consortium [2001, 2004]
URL	http://www.expasy.org/sprot; http://www.geneontology.org/
Description	Swiss-Prot is a curated protein sequence database which strives to provide a high level of annotation, such as the function of a protein, its domain structure, post-translational modifications, variants, etc. It contains a minimal level of redundancy and high level of integration with other databases, including the PDB. The Gene Ontology project is a controlled vocabulary for annotation of proteins. The GO terms, elements of this vocabulary, are arranged in a network, allowing complex semantic relationships to be described. The BioSQL project has created a very detailed data model to contain both the Swiss-Prot and GO data. It offers parsers that can directly use the database files for both blocks of data.
Data source ID	biosql
Requires	-
DB schema	biosql
DB tables	31 tables in the biosql schema
Cross-references	The references in the Swiss-Prot database suffer from the drawback that only entire PDB entries but not chains are being referenced. Tracking these cross-references down to individual protein chains is not trivial, since given protein sequences may differ between Swiss-Prot and PDB entries. The links between PDB and Swiss-Prot in Columba are established by the PDB-SprotEC module (see A.2.11)
Primary data	5 ASCII files (585 MB)
Updates	weekly
Parser	BioPerl-DB
Available on website	yes
Module author	Raphael A. Bauer, Kristian Rother

A.2.3 Boehringer

Name	Biochemical Pathways - Cellular and Molecular Processes Michal [1993]
URL	http://us.expasy.org/tools/pathways
Description	The famous Boehringer-Mannheim-posters by Gerard Michal contain lots of biochemical reactions and pathways. They are also available online as HTML pages. From these, the positions of enzyme E.C. numbers were stored, allowing to generate hyperlinks to the map locations where particular enzymes occur.
Data source ID	boehringer
Requires	-
DB schema	columba
DB tables	boehringer
Cross-references	The Boehringer data is connected to PDB compounds via the four-digit E.C. numbers.
Primary data	HTML imagemaps (15 MB)
Updates	Never (Boehringer-Mannheim is now Merck)
Parser	Selfmade Python parser
Available on website	Yes
Module author	Kristian Rother

A.2.4 CATH

Name	CATH - Protein Structure Classification Orengo et al. [1997], recent changes in Pearl et al. [2005]
URL	http://www.biochem.ucl.ac.uk/bsm/cath
Description	CATH is a hierarchical classification of protein domain structures, which clusters proteins at four major levels, C lass, A rchitecture, T opology and H omologous superfamily. The classification was done manually, aided by several structure comparison procedures. Additionally, CATH contains a clustering by sequence homology on different sequence identity levels for proteins having the same topology. In Columba, all four levels of the CATH hierarchy and the homologous families are represented in a single table in an unnormalized form. A second table links the CATH entries to PDB chains. CATH also defines protein domains based on the structure. These domains often consist of multiple fragments of neighboring polypeptide chains. Owing to the complexity that an aproppriate representation would impose on the database, they were not included in Columba.
Data source ID	cath
Requires	pdb
DB schema	columba
DB tables	cath, cath_chain_link
Cross-references	The CATH data is linked to individual PDB chains.
Primary data	Two ASCII tables (2 MB)
Updates	Approximately once a year
Parser	Selfmade Python parser
Available on website	Yes
Module author	Stefan Günther

A.2.5 DSSP

Name	DSSP - Definition of Secondary Structure of Proteins Kabsch and Sander [1983]
URL	http://www.cmbi.kun.nl/gv/dssp
Description	Secondary structures are calculated from the backbone H-bonds within protein structures using the DSSP program. The DSSP classification has been the de facto standard for secondary structure classification for many years. It defines seven secondary structure types, namely H elix, G (3_{10}-Helix), I 5-helix, E xtended (or beta sheet), hydrogen bonded T urn, B residue in isolated beta-bridge and S bend. In Columba, sequences of secondary structure codes for each polypeptide chain are stored. It is enforced that they have exactly the same length as the chain sequences from the PDB datasource.
Data source ID	dssp
Requires	pdb
DB schema	columba
DB tables	dssp
Cross-references	Each entry in the DSSP table is linked to the PDB chain it was calculated from.
Primary data	ASCII files in DSSP format (about 2 GB)
Updates	Can be calculated as soon as new PDB entries arrive.
Parser	Selfmade Python parser
Available on website	Yes
Module author	Kristian Rother

A.2.6 ENZYME

Name	ENZYME Bairoch [2000]
URL	http://www.expasy.org/enzyme
Description	These are the official definitions of enzyme names and numbers. The enzyme classification (E.C.) numbers classify an enzymatic function chemically. The ENZYME database contains names, descriptions, synonyms, sequence references, catalyzed reactions and related diseases. The enzyme names along with most of the descriptions are stored in one table in Columba. A small second table links all the synonyms with the canonical names of the enzymes.
Data source ID	enzyme
Requires	-
DB schema	columba
DB tables	enzyme, enzyme_name
Cross-references	For ENZYME, references were created by matching the four-number enzyme classification (E.C. numbers) with E.C. numbers given in the PDB compound description.
Primary data	Single ASCII file (2 MB)
Updates	Released about four times a year.
Parser	BioPython parser (Bio.ENZYME)
Available on website	Yes
Module author	Kristian Rother

A.2.7 Gene Ontology Annotation

Name	GOA - Gene Ontology Annotation Camon et al. [2004]
URL	http://www.ebi.ac.uk/goa
Description	GOA (GO Annotation@EBI) is a project run by the European Bioinformatics Institute that aims to provide assignments of gene products to the Gene Ontology (GO) resource. In the GOA project, this vocabulary will be applied to a non-redundant set of proteins described in the UniProt (Swiss-Prot/TrEMBL) and Ensembl sequence databases. The annotation is done automatically, and it is known that the assignments within GOA are not always aproppriate. However, it is the only resource of this kind. The single GOA table used in Columba contains identifiers of GO terms and associated Swiss-Prot entries.
Data source ID	goa
Requires	biosql
DB schema	goa
DB tables	goatable
Cross-references	Each annotation in the GOA data references a Swiss-Prot entry.
Primary data	ASCII file (406 MB)
Updates	About monthly
Parser	Uses COPY to load data into the database directly.
Available on website	Yes
Module author	Raphael A. Bauer

A.2.8 KEGG

Name	KEGG - Kyoto Encyclopedia of Genes and Genomes Kanehisa et al. [2004]
URL	http://www.genome.ad.jp/kegg
Description	KEGG contains integrated and interconnected data on genes, genomes, organisms, DNA and protein sequences, enzymes - and pathways. Among the pathways, there are metabolic and signal transduction pathways and some other schemas describing general biological processes in the cell, e.g. translation. In Columba, only the metabolic pathway data was included. It consists of lists of E.C. numbers of enzymes that group to pathways. Each enzyme can belong to one or more pathways. The relationships between enzymes (reactions, etc.) are not represented in Columba yet.
Data source ID	kegg
Requires	-
DB schema	columba
DB tables	kegg_enzyme, kegg_pathway, kegg_pe_link
Cross-references	For KEGG, references were created by matching the four-number enzyme classification (E.C. numbers) with E.C. numbers given in the PDB compound description.
Primary data	HTML files (about 5 MB).
Updates	Single pathways are modified about once a week.
Parser	Selfmade Python parser
Available on website	Yes
Module author	Kristian Rother

A.2.9 NCBI Taxonomy

Name	NCBI Taxonomy Wheeler et al. [2000]
URL	http://www.ncbi.nlm.nih.gov/Taxonomy/
Description	The NCBI taxonomy database contains the names of all organisms from the gene and protein databases with at least one nucleotide or protein sequence. The organisms are organized in a phylogenetic hierarchy, including all the higher-order nodes. The BioSQL project provides a parser that automatically downloads the taxonomy files and inserts them into the BioSQL data model that also fits the Swiss-Prot and GO data. However, both parsers applied to the same tables will mix the NCBI taxonomy data with data from the Swiss-Prot. Therefore, a separate schema with the BioSQL tables was defined for the NCBI taxonomy within Columba.
Data source ID	ncbi
Requires	-
DB schema	ncbi
DB tables	31 tables equivalent to those in the biosql schema. Most of them are empty, though, because only the taxonomic part of the tables is used further.
Cross-references	NCBI does not contain references on its own, but each Swiss-Prot entry contains an organism name that is identical to an entry in the NCBI taxonomy. Using this information, the ncbi_mapping (A.3.19) module connects the taxonomy to chains from the PDB.
Primary data	ASCII files (30 MB)
Updates	About monthly
Parser	BioPerl-DB
Available on website	Yes
Module author	Raphael A. Bauer

A.2.10 PDB Clusters

Name	PDB Clusters Li et al. [2001]
URL	ftp.rcsb.org/pub/pdb/derived_data/NR
Description	Contains weekly updated lists of clusters of PDB chains by sequence identity. There are three clusterings by 50,70 and 90 percent sequence identity available.
Data source ID	pdb_clusters
Requires	pdb
DB schema	columba
DB tables	pdb_clusters
Cross-references	Each entry of a cluster directly references one PDB chain.
Primary data	three ASCII files (5 MB)
Updates	weekly
Parser	Selfmade Python parser
Available on website	Yes
Module author	Kristian Rother

A.2.11 PDBSprotEC

Name	PDBSprotEC Martin [2004]
URL	http://www.bioinf.org.uk/pdbsprotec
Description	List of references from Swiss-Prot entries to PDB entries and ENZYME E.C. numbers. This list is used by biosql_mapping to establish links between tables from biosql and pdb.
Data source ID	pdbsprotec
Requires	pdb, biosql_mapping, enzyme
DB schema	columba
DB tables	pdbsprotec, chain_swissprot_link
Cross-references	PDBSprotEC contains references from Swiss-Prot entries to the PDB and ENZYME. They not only outnumber the links given in Swiss-Prot itself, but also reference individual PDB chains. This data is used to connect the Swiss-Prot data mapped by biosql_mapping (A.2.15) to the Columba schema with chains from the PDB. These Swiss-Prot-PDB links are also used as intermediate information for connecting PDB entries to the NCBI Taxonomy (A.3.19) and Gene Ontology Annotation (A.3.18).
Primary data	one ASCII file (3.5 MB)
Updates	weekly
Parser	Selfmade Python parser
Available on website	Yes
Module author	Kristian Rother

A.2.12 PISCES

Name	PISCES (Protein Sequence Culling Server) Wang and Dunbrack Jr [2003]
URL	dunbrack.fccc.edu/PISCES.php
Description	Contains a large number of culled subsets of the PDB. They are calculated as non-redundant lists on the PISCES server. Each list contains a number of PDB chains. The lists are composed using different sequence identity thresholds, maximum crystallographic resolutions, and allowing NMR structures or not. All PISCES lists were included in Columba. One table contains the names of the lists, a second one links the lists to PDB chains.
Data source ID	named_list
Requires	pdb
DB schema	columba
DB tables	named_list, list_entry
Cross-references	Each entry in a PISCES list references one chain from the PDB.
Primary data	Many ASCII files (12 MB)
Updates	Weekly
Parser	Selfmade Python parser
Available on website	Yes
Module author	Kristian Rother

A.2.13 PTGL

Name	PTGL - Protein Topology Graph Library May et al. [2004]
URL	http://sanaga.tfh-berlin.de/∼ptgl/ptgl.html
Description	The neighborhood relationships between secondary structural elements from most PDB structures are represented in the PTGL database. The secon_dat table contains the location and type of all secondary structures. The other three tables reference pairs of secondary structures in close contact, depending on the type of secondary structure (alpha helix only, beta sheet only, or both). The resulting topology graphs are used by the PTGL to visualize the arrangement of secondary structury, and to compare different topologies.
Data source ID	ptgl
Requires	pdb
DB schema	ptgl
DB tables	secon_dat, alpha, beta, albe
Cross-references	The PTGL data directly references PDB chains.
Primary data	ASCII files (50 MB)
Updates	about every two months
Parser	Selfmade Python parser
Available on website	No
Module author	Kristian Rother and Patrick May

A.2.14 SCOP

Name	SCOP - Structural Classification of Proteins Murzin et al. [1995], recent changes in Andreeva et al. [2004]
URL	http://scop.berkeley.edu
Description	SCOP is a hierarchical classification of protein folds and families. Most structures in the PDB have been analyzed manually and assigned a class, a fold, a superfamily, a family and, where aproppriate, domains. The class roughly describes what types of secondary structure dominate in a protein. The fold is a description of the arrangement of secondary structures not necessarily implying evolutionary relationship. Superfamilies are groups of proteins that usually are at least distant relatives, and families are closely related. All four levels of hierarchy are stored within Columba in a single table (not normalized). The domains and their boundaries (residue numbers) are also deposited. A second table links SCOP entries to chains from the PDB. As the annotation is done by experts in the SCOP team, the data is generally of high quality, but grows slowly.
Data source ID	scop
Requires	pdb
DB schema	columba
DB tables	scop, chain_scop_link
Cross-references	The SCOP database contains direct references from each SCOP domain to one chain from the PDB. At time of writing, 47 references from SCOP primary data were out-of-date because the corresponding PDB structures had become obsolete in the PDB.
Primary data	Three ASCII tables (11.5 MB)
Updates	Once a year
Parser	BioPython parser (Bio.SCOP)
Available on website	Yes
Module author	Kristian Rother

A.2.15 SYSTERS

Name	SYSTERS - A Protein Family Webserver Krause et al. [2000]
URL	http://systers.molgen.mpg.de
Description	SYSTERS is a large pool of protein sequences assigned to families. The clustering algorithm is based on massive Smith-Waterman-alignments performed on a parallel computer. However, SYSTERS does not use a fixed sequence identity threshold to define families. The threshold depends on how far sequence clusters are apart from other clusters. The SYSTERS website offers protein families calculated from a large set of sequence databases. For Columba, only the sequences from SYSTERS 3.2 (late 2003) referencing the PDB were available. These resemble mostly an earlier release of the Swiss-Prot. The families are stored in a single table, connecting PDB chains to integer values indicating the SYSTERS protein families.
Data source ID	systers
Requires	pdb
DB schema	columba
DB tables	chain_systers_link
Cross-references	The Systers data contains direct references to PDB chains.
Primary data	ASCII dump (8 MB)
Updates	Once a year
Parser	Selfmade Python parser
Available on website	No
Module author	Kristian Rother

A.3.16 BioSQL Mapping

Name	Mapping of biosql to the columba schema
URL	-
Description	Maps information from the biosql schema into more concise tables in the columba schema. The mapping is done as a materialized view, selecting data from multiple tables and storing it in a single new one. After mapping the Swiss-Prot and GO data, the data from PDBSprotEC is used to establish direct links to the PDB tables.
Data source ID	biosql_mapping
Requires	biosql
DB schema	columba
DB tables	swissprot_entry, swissprot_dbref, swissprot_reference, swissprot_reference_link, go_term, go_relationship, go_synonym, go_*
Cross-references	References between PDB and Swiss-Prot entries could be taken from the original Swiss-Prot entries. But these only annotate entries, not chains. The PDBSprotEC database (A.2.11) contains carefully curated links to the Swiss-Prot database. These links are used by biosql_mapping to establish chainwise references between the *chain* and *swissprot_entry* table.
Available on website	Yes
Module author	Raphael A. Bauer, Kristian Rother, Silke Trissl

A.3.17 Full text search

Name	Full text search
URL	-
Description	Creates the full text search indices from text fields in many tables throughout the columba schema. For this, a huge view bringing most tables together is used. The search is performed by the tsearch2 module that can be added to the PostGreS RDBMS. It provides special data types indexing string-type columns.
Data source ID	fts
Requires	scop,cath,kegg,boehringer,enzyme, ncbi_mapping,goa_mapping.
DB schema	public
DB tables	total_table
Cross-references	-
Available on website	Yes
Module author	Raphael A. Bauer

A.3.18 GOA Mapping

Name	Mapping of goa to the columba schema
URL	-
Description	Creates links from PDB chains to GO terms using the annotation provided by GOA. The chain_go_link table contains references from GO terms to PDB chains. The go_graph_path table represents the hierarchical structure between the GO terms, and the gotype table groups the GO terms to the three classes 'molecular function', 'biological process' and 'cellular component'.
Data source ID	goa_mapping
Requires	biosql,goa
DB schema	columba
DB tables	chain_go_link, go_graph_path, gotype
Cross-references	Uses the references from GOA to Swiss-Prot entries and references from Swiss-Prot to PDB chains calculated by biosql_mapping to link the GO terms to individual chains in the PDB. Also, GO terms and their parent nodes are indexed in the go_graph_path table to allow queries involving related terms.
Available on website	Yes
Module author	Raphael A. Bauer, Kristian Rother, Silke Trissl

A.3.19 NCBI Mapping

Name	Mapping of ncbi to the columba schema
URL	-
Description	Maps information from the ncbi schema into more concise tables in the columba schema. Links the mapped Swiss-Prot entries to PDB entries using the links calculated by biosql_mapping. The ncbi_taxonomy table contains entries on all organisms, the ncbi_chain_link table links them to PDB chains, and the tax_graph_path table represents the hierarchical structure of the taxonomic tree.
Data source ID	ncbi_mapping
Requires	ncbi, biosql_mapping
DB schema	columba
DB tables	ncbi_chain_link, ncbi_taxonomy, tax_graph_path
Cross-references	Uses the links from Swiss-Prot to PDB created by biosql_mapping to map NCBI taxonomy entries - that are referenced by Swiss-Prot entries - on individual PDB chains. Also calculates references between each node in the taxonomy tree to each of its parent nodes to allow queries across entire branches of the taxonomy.
Available on website	Yes
Module author	Raphael A. Bauer, Kristian Rother, Silke Trissl

A.3.20 Web site data structures

Name	Web site data structures
URL	-
Description	An individual data source was implemented to prepare the database for web user sessions. It creates data structures containing data for each session, including session-ID's, queries, query results, and a few database views and functions. It requires the full text search to be initialized beforehand. Effectively, the Columba database can be accessed by the web interface immediately after the web data source has been processed.
Data source ID	web
Requires	fts.
DB schema	web
DB tables	loginfo,internal_id,queryinfo
Cross-references	-
Available on website	Yes
Module author	Silke Trissl, Raphael A. Bauer

B Distance matrix from the comparison of databases

On the next page, the overlap matrix is shown, by which the UPGMA tree in figure 9.5 was calculated.

	BRE	CAT	CE	DAL	ENZ	GO	HOM	HSS	IPRO	KEG	NCB	PDB	PSE	SCP	SWS	UNI
BRENDA	100	46	46	45	84	45	35	43	44	52	45	40	45	46	48	48
CATH	46	100	83	82	42	67	54	66	71	25	67	62	67	82	73	73
CE	46	83	100	93	42	67	57	67	76	25	67	64	67	85	74	74
DALI	45	82	93	100	41	67	56	68	76	25	67	64	67	85	74	74
ENZYME	84	42	42	41	100	46	31	47	40	57	46	45	46	43	45	45
GO	45	67	67	67	46	100	44	79	64	26	99	76	99	76	88	88
HOMSTRAD	35	54	57	56	31	44	100	41	51	20	44	38	44	51	48	48
HSSP	43	66	67	68	47	79	41	100	58	27	80	94	80	78	78	78
INTERPRO	44	71	76	76	40	64	51	58	100	26	64	54	64	73	68	68
KEGG	52	25	25	25	57	26	20	27	26	100	26	25	26	25	26	26
NCBI	45	67	67	67	46	99	44	80	64	26	100	76	100	76	89	89
PDB	40	62	64	64	45	76	38	94	54	25	76	100	76	75	74	74
PDBSPROTEC	45	67	67	67	46	99	44	80	64	26	100	76	100	76	89	89
SCOP	46	82	85	85	43	76	51	78	73	25	76	75	76	100	83	83
SWISS	48	73	74	74	45	88	48	78	68	26	89	74	89	83	100	100
UNIPROT	48	73	74	74	45	88	48	78	68	26	89	74	89	83	100	100

Table B.1: Matrix of the overlap between pairs of datasets linking to the PDB. Each value was calculated as the number of entries at the intersection of both sets divided by the number of entries in their union. The numbers are given in percent. For instance, 84% of the PDB entries annotated by either BRENDA or ENZYME, are referenced by both databases.

C Dataset used for the packing analysis of protein ligands

The PDB-codes of protein structures containing at least one from the 6 coenzymes and 7 small organic molecules are listed below. The dataset was created using the procedure described in table 8.3.

coenzyme and small organic ligand-binding proteins											
119l	19hc	1a2q	1a44	1a4i	1a6m	1a7p	1a7s	1a8p	1ag9	1ai2	1aii
1amt	1aoe	1aok	1aoq	1apv	1aq0	1ars	1aru	1at5	1atg	1ayf	1azo
1b0b	1b5p	1b5q	1b63	1b6g	1b93	1b9h	1bbh	1bdb	1be0	1bel	1bf6
1bff	1bg2	1bgp	1bhp	1bif	1bio	1bmd	1bn7	1bqu	1bsg	1bt0	1btc
1bu5	1bu7	1bw9	1bxq	1byf	1c0p	1c1d	1c3d	1c52	1c53	1c5e	1c75
1c7n	1caz	1cb8	1cbn	1cg5	1cgd	1cgo	1cjc	1cll	1co6	1cot	1cpo
1cpq	1cqx	1cru	1cs3	1cs6	1ct5	1ctj	1cv8	1cxc	1cxp	1cxq	1cxy
1cy5	1cyd	1cyo	1czh	1d0c	1d1q	1d2n	1d3b	1d3g	1d4a	1d4o	1d7o
1d7p	1d8d	1daa	1dbf	1dci	1df7	1dgf	1dgm	1dje	1djl	1djr	1dk0
1dk8	1dl2	1dlj	1dll	1dlw	1dly	1dnc	1dnl	1dnu	1dqa	1dqt	1dry
1ds0	1dss	1dug	1dw0	1dxy	1dy5	1dyr	1dz4	1e0c	1e18	1e1m	1e2v
1e39	1e42	1e4c	1e4h	1e4m	1e6b	1e6u	1e6w	1e7w	1e87	1e93	1e9v
1eb6	1eca	1ecv	1edu	1eem	1een	1eg5	1ej2	1ek6	1ekf	1ekj	1el5
1ela	1elq	1emd	1enp	1eon	1eq2	1esw	1eua	1euw	1ew0	1ew6	1eyn
1ez1	1f0v	1f0x	1f0y	1f20	1f2d	1f3a	1f4g	1f4p	1f5v	1f74	1f7t
1f8g	1f8r	1f9y	1fc4	1fcj	1fdr	1fec	1fiu	1fj0	1fjq	1fjs	1fju
1fk5	1fk8	1fl2	1flj	1flm	1fmc	1fnd	1fni	1foa	1fqt	1fr9	1frb
1fs7	1ft5	1fv0	1fw1	1fy4	1fz1	1fz7	1g0c	1g0o	1g2w	1g3m	1g42
1g4k	1g55	1g5h	1g66	1g6s	1g6x	1g79	1g7a	1g87	1g8f	1g8i	1g8k
1g8l	1g9r	1ga2	1ga6	1gci	1gcv	1gd1	1gde	1gdv	1gee	1geg	1get
1gew	1giq	1gk8	1gk9	1gkl	1gkm	1gmx	1gnl	1gof	1goi	1gox	1gpe
1gqv	1gr0	1grb	1gre	1gsa	1gse	1gsk	1gte	1gtv	1gu1	1gu7	1gui
1gve	1gvf	1gvz	1gwd	1gwe	1gwh	1gwi	1gwm	1gwu	1gxm	1gxy	1gy2
1gy8	1gyo	1gz7	1gzf	1gzw	1h01	1h0d	1h0s	1h1a	1h2e	1h32	1h41
1h4p	1h50	1h5b	1h5q	1h5u	1h5y	1h6h	1h7c	1h80	1h8e	1h8u	1h97
1ha3	1hbn	1hcz	1hdo	1he4	1he7	1heu	1hf6	1hj9	1hjx	1hnl	1hoz
1hs6	1ht6	1htw	1hvy	1hw3	1hx0	1hxp	1hye	1hyo	1hz4	1i0d	1i0r
1i0s	1i13	1i19	1i1w	1i24	1i2k	1i36	1i3k	1i58	1i6l	1i8f	1ia7
1iat	1icp	1icr	1idr	1ie0	1ifr	1ihg	1iho	1ijn	1iju	1in4	1io7
1iom	1iqc	1ird	1iso	1isp	1it2	1itx	1iuo	1iuq	1ivn	1iw0	1iwh
1iy8	1iyh	1iyn	1j05	1j0p	1j18	1j31	1j3b	1j3v	1j4r	1j54	1j58
1j5p	1j71	1j77	1ja1	1ja9	1jak	1jay	1jb9	1jbe	1jbm	1jbz	1jcz
1jdr	1jdw	1jfb	1jgt	1jip	1jjt	1jk7	1jkv	1jlj	1jlv	1jni	1jnr
1jo0	1jom	1jou	1jp4	1jpn	1jpz	1jq5	1jr8	1jrr	1jtv	1ju2	1jub
1jue	1juh	1juv	1jv0	1jyd	1jye	1k07	1k0e	1k0n	1k1e	1k20	1k2e
1k38	1k3i	1k3x	1k3y	1k4m	1k4o	1k55	1k5c	1k5n	1k66	1k6x	1k77
1k87	1k96	1ka1	1kae	1kb0	1kcz	1kea	1kep	1kew	1kfw	1kg2	1kgd
1khb	1kjq	1kl1	1kli	1kmj	1kmv	1knm	1ko3	1kol	1kop	1kq1	1kq6
1kqc	1kqf	1kqp	1kqr	1kqy	1kr7	1krh	1kta	1ktb	1kv5	1kv9	1kwg

coenzyme and small organic ligand-binding proteins										
1kx9	1kzk	1l1g	1l5w	1l5x	1l6r	1l6w	1l6y	1l7e	1l7f	1l8n	1l8s
1l9l	1l9x	1lb3	1lbk	1lc3	1ld8	1lfk	1lg5	1li1	1lj5	1lj8	1ljo
1lk9	1lka	1lkc	1ll3	1lld	1llp	1lm4	1lni	1lo7	1lqa	1lqt	1lqu
1lu0	1lua	1luc	1lug	1luq	1lvw	1lw4	1ly3	1lyq	1m0u	1m1q	1m2t
1m2x	1m33	1m3k	1m5e	1m5q	1m6i	1m7s	1m7v	1m85	1m8z	1m9h	1mb4
1mba	1mdx	1meg	1mej	1mg5	1mi3	1mj4	1mju	1mk0	1mki	1mkz	1mn1
1mo0	1mo9	1mog	1mpg	1mpx	1mqd	1mqv	1mr3	1mrq	1msk	1mu7	1mus
1mv8	1mvl	1mwv	1mx3	1mxd	1mxg	1myt	1mz4	1n0w	1n0x	1n0y	1n2e
1n2r	1n2s	1n3l	1n40	1n45	1n4w	1n55	1n5n	1n5u	1n62	1n6h	1n7h
1n82	1n97	1n9g	1n9l	1na0	1nb9	1nc7	1ndg	1ne2	1nf5	1nf9	1nff
1nfp	1nh0	1nhc	1nhp	1njh	1nln	1nm2	1nng	1no5	1nof	1nox	1np7
1nrg	1nrj	1ntf	1nuu	1nvi	1nvk	1nvm	1nw2	1nxd	1nxw	1nyt	1nza
1nzk	1nzn	1nzo	1nzy	1o04	1o0s	1o26	1o2d	1o4s	1o61	1o66	1o69
1o6z	1o7e	1o7j	1o7n	1o7q	1o82	1o8a	1o94	1o97	1o9r	1oa1	1oaa
1oae	1oaf	1oao	1obb	1obo	1obz	1oc2	1oc7	1ocj	1od3	1odk	1odo
1odt	1oe4	1oe8	1oen	1oew	1of8	1ofc	1ofd	1ofw	1ogi	1ogq	1oh0
1oh4	1ohb	1ohl	1oht	1oi2	1oi6	1oiv	1ojh	1ojk	1ojr	1ok0	1okt
1ol0	1oll	1om0	1omy	1onw	1oo0	1oof	1ooh	1opk	1oq1	1oqc	1oqu
1oqv	1or0	1orb	1orr	1os6	1osf	1ouw	1ow4	1owl	1oyc	1oyg	1oyj
1ozn	1p0f	1p0i	1p1h	1p1j	1p36	1p4c	1p4k	1p4m	1p4n	1p6o	1p77
1p7g	1p80	1p90	1p9g	1pb0	1pb1	1pbe	1pbt	1pby	1pda	1pfb	1pfz
1pg4	1pj5	1pj9	1pjc	1pjx	1pkh	1pkw	1pl3	1pl8	1pm1	1pmh	1pmj
1pmm	1pn0	1pn2	1po5	1pp0	1ppd	1ppn	1pqu	1pt6	1pt7	1pu6	1pwa
1pwm	1pxg	1pyf	1pz3	1pzh	1q0m	1q0q	1q16	1q1c	1q1f	1q1r	1q25
1q35	1q40	1q4f	1q4g	1q4u	1q5d	1q5y	1q7r	1q8f	1q92	1q9i	1qa7
1qb0	1qd1	1qd9	1qfm	1qfy	1qfz	1qgj	1qgq	1qh3	1qh4	1qh5	1qh8
1qip	1qj4	1qj5	1qks	1qkw	1qmq	1qmy	1qnr	1qop	1qow	1qpc	1qq5
1qqu	1qsa	1qsg	1qus	1qv0	1qvw	1qwl	1qwm	1qwr	1qx4	1qxm	1qxo
1qxy	1qy0	1qyf	1qyw	1qz9	1r0k	1r0u	1r2q	1r3v	1r4p	1r6d	1r6u
1r75	1r85	1r8s	1r9d	1r9o	1r9z	1ra3	1ra6	1ra9	1rcq	1rdq	1req
1rey	1rfx	1rg8	1rgz	1rhc	1rhf	1rii	1rjd	1rk6	1rku	1rkx	1rlj
1rlk	1rm4	1rn8	1rnq	1rrv	1rtm	1rtz	1ruk	1rwh	1rwj	1rwy	1rx0
1rxy	1rz3	1rzf	1rzh	1s0k	1s1d	1s1p	1s1q	1s22	1s3e	1s3w	1s44
1s5a	1s5d	1s5u	1s65	1s69	1s81	1s99	1sac	1sbz	1sck	1scw	1sdi
1sdu	1sdw	1sff	1sft	1sfx	1sg6	1sht	1sii	1sk7	1skg	1slu	1sn4
1snr	1sny	1sq2	1sr7	1ss4	1ssx	1st9	1sug	1sxr	1sz8	1szd	1szh
1szn	1t0i	1t1v	1t2d	1t2e	1t3i	1t3q	1t56	1t5b	1t61	1t6e	1t6g
1t7m	1t7q	1t8h	1t8q	1ta3	1tbb	1tbf	1tcs	1tdz	1tgt	1thq	1ti7
1tj9	1tjg	1tn3	1toa	1tpz	1tqi	1tr0	1trb	1tsd	1tt2	1tu1	1tu9
1tuh	1twi	1tz0	1tzb	1u02	1u1i	1u55	1u60	1u7o	1u84	1u98	1uar
1uas	1ubp	1udc	1ued	1uet	1uf5	1ug6	1ui9	1uis	1ul3	1um0	1umm
1uow	1uq5	1urn	1urr	1us0	1us3	1us5	1usc	1usf	1usp	1ut1	1uu3
1uu5	1uuj	1uuq	1uuy	1uvq	1uw4	1uwk	1uws	1uxa	1uxj	1uxy	1uy4
1uyp	1uyz	1uzi	1v03	1v0n	1v1r	1v2d	1v3v	1v3w	1v4v	1v4x	1v5v
1v6s	1v8c	1v8f	1v93	1v97	1v9m	1v9y	1va4	1vc4	1vdw	1vfh	1vfr
1vgi	1vhn	1vhq	1vht	1vi3	1vi9	1vim	1vjf	1vjo	1vjp	1vk5	1vkb
1vkh	1vki	1vkm	1vkp	1vl7	1vlg	1vlj	1vlp	1vlr	1vly	1vm9	1vmb
1vme	1vmf	1vp4	1vp8	1vrk	1vyb	1vyi	1vyr	1vzo	1w0p	1w1o	1w2i
1w32	1w3i	1w3o	1w4x	1w53	1w6n	1w8o	1wab	1wad	1wei	1wej	1wmd
1wr8	1x8q	1xff	1xfj	1xfp	1xg5	1xqf	1ycc	1ygh	1yve	1zrn	256b
2aac	2ae2	2cbc	2ccy	2cst	2cy3	2erl	2fcr	2gdm	2gpa	2hbg	2hmz
2hts	2izk	2lhb	2lyo	2nad	2nsy	2olb	2pvb	2rti	3c2c	3cao	3cyr
3dfr	3eug	3gcb	3grs	3lzt	3nul	3pcc	3sdh	3sil	451c	4fgf	5cp4
5nul	7a3h	7aat	7est	7odc	8a3h	8gss	9ldt				

D Web links

database	URL
Boehringer-Mannheim Biochemical Pathways	us.expasy.org/tools/pathways
BRENDA enzyme information system	www.brenda.uni-koeln.de
CATH protein structure classification	www.biochem.ucl.ac.uk/bsm/cath
Combinatorial Extension structure alignment (CE)	cl.sdsc.edu/ce.html
DALI domain directory	www.ebi.ac.uk/dali
DSSP software	www.cmbi.kun.nl/gv/dssp
Enzyme nomenclature database (ENZYME)	www.expasy.org/enzyme
Gene Ontology Terms (GO)	www.geneontology.org
Gene Ontology Annotation (GOA)	www.ebi.ac.uk/goa
Homologous Structure Alignments (HOMSTRAD)	www-cryst.bioc.cam.ac.uk/∼homstrad
Homology-derived Secondary Structure of Proteins (HSSP)	swift.cmbi.kun.nl/swift/hssp
InterPro database	www.ebi.ac.uk/interpro
Kyoto Encyclopedia of Genes and Genomes (KEGG)	www.genome.ad.jp/kegg
NCBI Taxonomy	www.ncbi.nlm.nih.gov/Taxonomy
Protein Data Bank (PDB)	www.pdb.org
PDB-clustering by cd-hit	ftp.rcsb.org/pub/pdb/derived_data/NR
PDBSprotEC database	www.bioinf.org.uk/pdbsprotec
PISCES non-redundant lists	dunbrack.fccc.edu/PISCES.php
PTGL	sanaga.tfh-berlin.de/ ptgl/ptgl.html
Structural Classification of Proteins (SCOP)	scop.berkeley.edu
Swiss-Prot protein knowledgebase	www.expasy.org/sprot
Systers	systers.molgen.mpg.de
Universal Protein Resource (UniProt)	www.ebi.ac.uk/uniprot

E Abbreviations and terms used

Apache	An Open Source web server. See http://www.apache.org.
CATH	C lass, A rchitecture, T opology and H omologous superfamily. Database of protein folding classification, half-automatically assigned hierarchical folding classes, available for many protein structures.
CGI	C ommon G ateway I nterface. An interface by which web servers, like Apache can interact with scripting languages (like Perl, PHP or Python) called from a web page.
CVS	C oncurrent V ersions S ystem. A software that helps multiple programmers working on the same project to maintain a consistent source code repository.
Compound	a biologically relevant unit of one or several protein chains. The two chains of an IgG antibody are forming a single compound, while an enzyme and its peptide inhibitor should be two separate compounds.
DSSP	D ictionary of S econdary S tructures in P roteins, software and file format to calculate secondary structures from a protein structure. The de facto standard procedure.
Data source	a program module that is responsible for retrieving data for one specific topic and creating entries with this data in the database.
Entity relationship model	term from computer science. It is a graphical notation for displaying data types and/or logical relationships in structured data.
function	In a relational database, this is a piece of procedural code that is not restricted to SQL queries (e.g. calculating a standard deviation for a column, or performing a sort operation).
GO terms	G ene O ntology terms. A controlled vocabulary of protein and gene function and location-related words.
GOA	G ene O ntology A nnotation. Huge automatic assignment of GO terms to Swiss-Prot entries.
index	Data structure in a relational database that is used to accelerate queries on a specific column of a table.
KEGG	K yoto E ncyclopedia of G enes and G enomes. Contains metabolic and regulatory pathways linked to genes from different organisms.
MESH	M E dical S ubject H eadings, database of precisely defined medical terms.
MMCif	Protein structure file format successor to the PDB format. Files are longer, but the information is both more detailed and structured unambiguously.
MSD	M acromolecular S tructure D atabase. A cleaned, structured PDB.
metadata	information that specifies the origin of an entry. The metadata tells when, by whom, by which application and using which parameters the entry has been created.
module	a logical unit of program code in the Python language represented by one source code file. Corresponds to class files in Java.
NMR	N uclear M agnetic R esonance spectroscopy, a method based on spin differences of atom nuclei used for 2D/3D-structure determination.
pathway	A collection of biochemical reactions that are connected with each other in a metabolism. Not necessarily linear, because side reactions may also be involved.
PD	p acking d ensity of an atom. For the definition, see formula 5.1.

PDB	the P rotein D ata B ank containinng all known 3D-structures of proteins. See http://www.pdb.org
Perl	a widely-used object-oriented scripting language. Similar to Python in many aspects. See http://www.perl.org
PostGreSQL	An open-source RDBMS. See www.postgres.org.
Python	an object-oriented, interpreted programming language. Very flexible and easy to maintain, but quite slow. See http://www.python.org
RDBMS	R elational D ata B ase M anagement S ystem. A software that allows relational databases to be constructed, filled with data and queried using SQL.
schema	A data structure within relational databases that holds a number of tables.
SCOP	S tructural C lassification O f P roteins, database of manually assigned hierarchical folding classes, available for most protein structures.
sequence	In a relational database, it is a variable that counts something - in most cases, the number of entries in a table.
SQL	S tructured Q uery L anguage, a language for database queries and data definitions.
Z_{rms}	z-score-rms according to Pontius et al. [1996]. Quantifies the deviation of atomic packing densities from reference values. Defined in formula 8.2.
use case	term from software engineering describing the scenarios, in which a program has to work.
view	data structure in a relational database that is used to format data from tables. It can be used like a table itself, providing a convenient way to access data.

F Additional software used

Columba

The Columba database was implemented on a **PostGreSQL 7.4** server, including the **tsearch2** module providing full text indexing capabilities. The application managing the database content was written in **Python 2.3**. Database access from Python programs was done with the **pygresql** module. Single data sources used parsers from the **BioPython** module library (SCOP, ENZYME, PDB), the **pyxml** XML library (InterPro) and **BioPerl** with the **BioPerl-DB/BioSQL** extension (BioSQL, GOA, NCBI). For handling PDB files, the **moltools** program suite by Andreas Hoppe was used. The Python modules for parsing PDB headers and maintaining a local PDB copy were contributed to the **BioPython** project. The Columba web site was implemented on a **Apache** web server using **mod-python** CGI scripting module. The web interface also required **pso** for session management. Web-based 3D-views of proteins were displayed by **JMol**. Other software necessary for maintaining Columba are **CVS**, **awk**, and **DSSP**. All software used for Columba (excluding DSSP) is available under the conditions of the GNU General Public License or similar Open-Source licenses.

For manual data analysis, the SQL interfaces **psql** and **Aqua Data Studio** were used. For automatic analyses, **Python** scripts accessing the database were written.

Protein packing

The atomic volumes and cavities have been calculated using a modified version of the **Contact** program by A.Goede Goede et al. [1997]. The packing densities and other derived features were performed by a number of **Java** and **Python** programs. The statistical tests were performed using the **R statistics package**. All molecular graphics were created with **PyMOL**.

Typesetting

This text was written in **LaTeX** using **pdflatex kpathsea 3.4.5**. Literature references were managed using **JabRef**. Most figures were created using either **XmGrace** or the **R statistics package** and processed using **GIMP**.

Curriculum vitae

Name: Kristian Rother

Academic degrees and positions

7/2002–9/2005	Graduate student at the Protein Structure Theory Group/Prof. C. Frömmel, Charite Berlin
10/1996–3/2002	Diploma of Biochemistry at the Freie Universität Berlin. Diploma thesis: Packungsdichte und Packungsdefekte in Proteinstrukturen (2002)
8/2001–2/2002	Web administrator at the Neuroinformatics Group/Prof. H. Herz, Institute of Biology, Humboldt Universität Berlin
5/2000–7/2001	Student coworker at the Protein Structure Theory Group/Prof. C. Frömmel, Charite Berlin
06/1996	Abitur at the 8.OG (Gymnasium Steglitz) Berlin

Skills

Native Languages	German, Finnish
Foreign Languages	English, Latin
System Languages	Python, Java, Pascal
Certificates	IBM DB2 database user

Web resources

1. Columba - Protein structure annotation database: **www.columba-db.de**
2. rtools for PyMOL - Protein structure visualization tools: **www.rubor.de/bioinf**

Exhibitions

1. Proteinstrukturkalligraphie, Lise-Meitner-Hörsaal, FU Berlin (2006)
2. Missing Link: Art Meets Biomedicine, The Rubelle & Norman Schafler Gallery, Brookly, N.Y. (2006)
3. arts & science in vitro - Der seltsame Tanz der sozialen Amöbe, public act at the Institute of Biochemistry, Charite Berlin (2006)
4. Missing Link - Communication in Arts and Science, BMM Charite Berlin (2005)

Publications

1. Günther S, **Rother K**, Frömmel C. Molecular flexibility in protein-DNA interactions. Biosystems. 2006 Feb 16; [Epub ahead of print].

2. **Rother K**, Dunkel M, Michalsky E, Trissl S, Goede A, Leser U, Preißner R. A structural keystone for drug design. Journal of Integrative Bioinformatics. 2006 Jan 19;0019. Online Journal.

3. **Rother K**, Michalsky E, Leser U. How well are protein structures annotated in secondary databases? Proteins. 2005 Sep 1;60(4):571-576.

4. Trissl S, **Rother K**, Müller H, Steinke T, Koch I, Preißner R, Frömmel C, Leser U. Columba: an integrated database of proteins, structures, and annotations. BMC Bioinformatics. 2005 Mar 31;6(1):81.

5. Hildebrand PW, **Rother K**, Goede A, Preißner R, Frömmel C. Molecular packing and packing defects in helical membrane proteins. Biophys J. 2005 Mar;88(3):1970-1977.

6. **Rother K**, Müller H, Trissl S, Koch I, Steinke T, Preißner R, Frömmel C, Leser, U. Columba: Multidimensional Data Integration of Protein Annotations. Lecture Notes in Computer Science. 2994 , 156-171. Springer, 2004.

7. **Rother K**, Preißner R, Goede A, Frömmel C. Inhomogeneous molecular density: reference packing densities and distribution of cavities within proteins. Bioinformatics. 2003 Nov 1;19(16):2112-2121.

8. Preißner R, Goede A, **Rother K**, Osterkamp F, Koert U, Frömmel C. Matching organic libraries with protein-substructures. J Comput Aided Mol Des. 2001 Sep;15(9):811-817.

9. Gille C, Goede A, Preißner R, **Rother K**, Frömmel C. Conservation of substructures in proteins: interfaces of secondary structural elements in proteasomal subunits. J Mol Biol. 2000 Jun 16;299(4):1147-1154.

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig ohne fremde Hilfe verfaßt und nur die angegebene Literatur und Hilfsmittel verwendet zu haben.

KristianRother
21. Mai 2006